

i Cover Page

Department of Computer Science

Examination paper for: TDT4237 (Software Security and data privacy) Spring 2023

Examination date: 15.05.2022

Examination time (from-to): 09:00-13:00

Permitted examination support material: NO support material is allowed

Answers can be done in English or Bokmål

Academic contact during examination: Per H. Meland

Phone: +4741108148

Academic contact present at the exam location: **NO**

OTHER INFORMATION

Get an overview of the question set before you start answering the questions.

Read the questions carefully, make your own assumptions and specify them in your answer. Only contact academic contact if you think there are errors or insufficiencies in the question set.

Make your own assumptions: If a question is unclear/vague, make your own assumptions and specify them in your answer. Only contact academic contact in case of errors or insufficiencies in the question set.

Notifications: If there is a need to send a message to the candidates during the exam (e.g. if there is an error in the question set), this will be done by sending a notification in Inspera. A dialogue box will appear. You can re-read the notification by clicking the bell icon in the top right-hand corner of the screen.

Weighting: The weight of each question is on the question. For **Closed Ended questions (1 point for each question if the answer is correct, 0 point if the answer is wrong)**. There is one main question that is worth 26 points and 14 open questions that are worth 4 points each.

Withdrawing from the exam: If you become ill during the exam or wish to submit a blank answer/withdraw from the exam for another reason, go to the menu in the top right-hand corner and click "Submit blank". This cannot be undone, even if the test is still open.

Access to your answers: After the exam, you can find your answers in the archive in Inspera. Be aware that it may take a working day until any hand-written material is available in the archive.

1 Case description (26 points)

Read the case description and tasks from the PDF and answer the 7 tasks below (use numbering):

Skriv ditt svar her

Format

B


I


U


\times_2


\times^2


$\frac{\square}{\square}$
























Σ



Words: 0

Maks poeng: 26

2/22

² Exposure of Sensitive Information (4 points)

```
import mysql.connector

class User:

    def __init__(self, username, password):
        self.username = username
        self.password = password
        self.db = mysql.connector.connect(
            host="localhost",
            user="yourusername",
            password="yourpassword",
            database="yourdatabase"
        )

# Store the username and password in the database
cursor = self.db.cursor()
sql = "INSERT INTO users (username, password) VALUES (%s, %s)"
val = (username, password)
cursor.execute(sql, val)
self.db.commit()
```

1) In this example, the code is vulnerable to sensitive data exposure because it stores the user's password in plain text. What measures should be taken to protect sensitive user data, such as passwords, and can you explain how user passwords should be protected in the database?

2) List one other example of vulnerability that can expose sensitive information such as passwords and explain briefly how the vulnerabilities could be exploited.

Skriv ditt svar her

Maks poeng: 4

3 CVSS for Heartbleed (4 points)

You discovered a vulnerability in the OpenSSL library and want to describe the severity of this using the Common Vulnerability Scoring System (CVSS).

An attack can be performed like this:

A successful attack requires only sending a specially crafted message to a web server running OpenSSL. The attacker constructs a malformed "heartbeat request" with a large field length and small payload size. The vulnerable server does not validate that the length of the payload against the provided field length and will return up to 64 kB of server memory to the attacker. It is likely that this memory was previously utilized by OpenSSL. Data returned may contain sensitive information such as encryption keys or user names and passwords that could be used by the attacker to launch further attacks.

Choose among the following metrics as input for the base score (you don't have to give the score itself) and write those below:

Attack vector: Network (N), Adjacent (A), Local (L), Physical (P)

Attack Complexity: Low (L), High (H)

Privileges required: None (N), Required (R)

Scope: Unchanged (U), Changed (C)

Confidentiality impact: High (H), Low (L), None (N)

Availability Impact: High (H), Low (L), None (N)

Integrity impact: High (H), Low (L), None (N)

Skriv ditt svar her

Format

B


I


U


~~x~~


x²


*I*_x





























Words: 0

Maks poeng: 4

4 Web Security (4 points)

<https://www.exampleTDT4237.com/reset-password?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTQyMjY3ODkwIiwiaWF0IjoxNjg3OTkxMjE5fQ>

Given an Example of a vulnerable link:

1) How can someone exploit this?

2) What can you do to protect such tokens? Name at least 2 ways.

Skriv ditt svar her

Maks poeng: 4

5 Insufficient logging and monitoring vulnerabilities (4 points)

```
LOGGING = {
    'version': 1,
    # Version of logging
    'disable_existing_loggers': True,
    #disable logging
    # Handlers #####
    'handlers': {
        'file': {
            'level': 'WARNING',
            'class': 'logging.FileHandler',
            'filename': 'dataflair-debug.log',
        },
        #####
        'console': {
            'class': 'logging.StreamHandler',
        },
    },
    # Loggers #####
    'loggers': {
        'django': {
            'handlers': ['file', 'console'],
            'level': 'LOGLEVEL',
        },
    },
}
```

The above code has insufficient security logging and monitoring vulnerabilities. Please propose a solution (lines that needs to change and new code) to fix the vulnerabilities. Explain what you have done.

Skriv ditt svar her

1	
---	--

Maks poeng: 4

6 Code to setup password policy (4 points)

Suppose the password policy of a system is as follows.

The password should be as long as possible and must contain at least 10 characters.

The passwords have to contain at least one character from the following four groups:

- **Upper-case letters: A–Z**
- **Lower-case letters: a–z**
- **Numbers: 0–9**
- **The following special characters: !#()+,.-=?@[_]{}-**

Spaces and the letters "æ", "ø" and "å" are not accepted

Your task is to develop code and configure Password Validators in Django to check the policy.

The following code partly takes care of the policy:

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
        'OPTIONS': {
            'min_length': 10,
        }
    },
    {
        'NAME': 'password_validators.validators.UppercaseValidator',
    },
    {
        'NAME': 'password_validators.validators.LowercaseValidator',
    },
    {
        'NAME': 'password_validators.validators.SymbolValidator',
    },
    {
        'NAME': 'password_validators.validators.NoNorValidator',
    },
]
```

```
class UppercaseValidator(object):
    def validate(self, password, user=None):
        if not re.findall('[A-Z]', password):
            raise ValidationError(
                _("The password must contain at least 1 uppercase letter, A-Z."),
                code='password_no_upper',
            )
```

```
class SymbolValidator(object):
    def validate(self, password, user=None):
        if not re.findall('[!@#$%^&* _-+=;:\'<>./?]', password):
            raise ValidationError(
                _("The password must contain at least 1 special character: " +
                  "[]{}|\`~!@#$%^&* _-+=;:\'<>./?"),
                code='password_no_symbol',
            )
```

Please add your extra code and explanation that takes care of the rest below.

Skriv ditt svar her

1	
---	--

Maks poeng: 4

7 Passwords (4 points)

Passwords is one of the most common ways to authenticate users, but it suffers from a number of weaknesses. Give three examples of such weaknesses. Further, give three examples of how password - based authentication can be improved.

Skriv ditt svar her

Format

B


I


U


\times_2


\times^2


\mathcal{I}_x
































Words: 0

Maks poeng: 4

8 OTP (4 points)

1. Explain encryption and decryption algorithm of One Time Pad (OTP).
2. Why is it considered to be "perfectly secure"?
3. Explain why it is insecure to use the same key to encrypt two or several messages using OTP.
4. What can an attacker do if he/she can manipulate the cipher text?

Skriv ditt svar her

Format

B


I


U


\times_2


\times^2


\int_x
























Σ



Words: 0

Maks poeng: 4

9 Biba and Bell-LaPadula (4 points)

How do the Biba and Bell-LaPadula models differ in their approach to enforcing data confidentiality and integrity?

Skriv ditt svar her

Format

B


I


U


\times_2


\times^2


\int_x
























Σ



Words: 0

Maks poeng: 4

¹⁰ Penetration Testing (4 points)

Explain the concept of Penetration Testing and its practical applications in modern-day cybersecurity. Provide examples of real-world scenarios where Penetration Testing can be utilized.

Skriv ditt svar her

Format

B


I


U


x_e


x^2


I_x
























Σ



Words: 0

Maks poeng: 4

¹¹ Social Engineering (4 points)

Mention at least four psychological factors that makes us susceptible to social engineering attacks.

Skriv ditt svar her

Maks poeng: 4

12 Malicious AI (4 points)

Your car uses a camera and AI to recognize road signs along the way. Mention at least four ways a malicious actor could attack the AI system?

Skriv ditt svar her

Format

B


I


U


x_2


x^2


I_x



























Σ



Words: 0

Maks poeng: 4

13 Spice (4 Points)

Salt and pepper are used to protect against what kind of attack?

Where do you usually store your salt?

When does the salt not work?

Where do you usually store your pepper?

Skriv ditt svar her

Format

B


I


U


x_2


x^2


I_x



























Σ



Words: 0

Maks poeng: 4

14 Fix vulnerability (4 points)

Consider the following code:









```
from django.db import connection
from django.http import HttpResponse
def my_view(request):
    # Get the user's input from the request
    user_input = request.GET.get('input')
    # Construct a raw SQL query
    query = "SELECT * FROM my_table WHERE column='%s'" % user_input
    # Execute the query
    cursor = connection.cursor()
    cursor.execute(query)
    # Process the results
    results = cursor.fetchall()
    # Return the results as an HTTP response
    return HttpResponse(results)
```


1. What is the main vulnerability here?
2. How can it be exploited (give example)?
3. Explain how it can be fixed
4. Write the code that fixes it (only the lines that need fixing)

Skriv ditt svar her

Format

B *I* U \times_2 \times^2 I_x

Σ 

Words: 0

Maks poeng: 4

¹⁵ Secure development practices (4 points)

You are put in charge of improving the software security of a new tech company. Briefly explain some of the practices you would want to recommend using (at least 4).

Skriv ditt svar her

Format

B


I


U


\times_e


\times^2


I_x
























Σ



Words: 0

Maks poeng: 4

¹⁶ Symmetric algorithm (1 point)

What is a symmetric algorithm and how can it be used in software to ensure secure data transmission?

Velg ett alternativ:

- A symmetric algorithm is a type of encryption algorithm that uses the same key for both encryption and decryption. In software, symmetric algorithms can be used to encrypt data before it is transmitted over a network, and then decrypt it on the receiving end using the same key. This ensures that the data is secure and cannot be intercepted or read by unauthorized parties.
- A symmetric algorithm is a type of compression algorithm that is used to reduce the size of data before it is transmitted over a network. In software, symmetric algorithms can be used to compress large files or data sets, making them easier and faster to transmit over a network. This can help to improve network performance and reduce bandwidth usage.
- A symmetric algorithm is a type of routing algorithm that is used to direct network traffic between different nodes. In software, symmetric algorithms can be used to optimize network routing and ensure that data is transmitted along the most efficient path. This can help to improve network performance and reduce latency.

Maks poeng: 1

17 Session hijacking (1 point)

A web developer is looking to mitigate the risk of session hijacking attacks on their website. Which of the following options would be effective in preventing session hijacking?

Velg ett alternativ:

- ☐ Setting the "HttpOnly" and "secure" flags on session cookies
- ☐ Deploying the website on a blockchain
- ☐ Signing the website certificate with a quantum-safe signature algorithm
- ☐ Enforcing a password policy of minimum 16 characters
- ☐ Ensuring that only AES is used to encrypt TLS traffic

Maks poeng: 1

18 Buffer overflow protection (1 point)

What are some recommended ways of defending against buffer overflow attacks?

Velg ett alternativ:

- ☐ Use parameterized queries, limit access to memory, escape special characters, sanitize all input and output.
- ☐ Use safe functions, leverage defences in compilers, use static analysis tools, rewrite in a type-safe language.
- ☐ Use non-standard C functions to manipulate strings, such as strcpy and strcat. These will not lead to buffer overflow vulnerabilities found in standard C functions, such as strncpy and strncat.
- ☐ Close all unused ports in the firewall, reduce the number of buffers, fine programmers making coding mistakes.

Maks poeng: 1

19 Security requirements (1 point)

What is a good security requirement?

Velg ett alternativ:

- ☐ Defining the choice of protection mechanism
- ☐ Stating what should be achieved, not how
- ☐ Stating what should not happen to the system
- ☐ A zero-knowledge proof
- ☐ A functional requirement

Maks poeng: 1

20 Mitigating Risks (1 Point)

In a scenario where a cyber attack has already compromised a company's database, which of the following countermeasures would provide the least effective return of investment (ROI) in terms of mitigating the damage caused by the breach?

Velg ett alternativ:

- ☐ Investing in incident response and digital forensics capabilities
- ☐ Implementing network segmentation and access controls
- ☐ Deploying endpoint protection solutions and intrusion detection systems
- ☐ Conducting regular vulnerability assessments and penetration testing
- ☐ Pursuing legal action against the attackers.

Maks poeng: 1

21 Vulnerabilities (1 point)

Consider the following code snippet:

```
from django.shortcuts import render
from django.http import HttpResponseRedirect
from django.urls import reverse
from .forms import ContactForm

def contact(request):
    if request.method == 'POST':
        form = ContactForm(request.POST)
        if form.is_valid():
            # Do something with the form data, like saving it to a database
            name = request.POST.get('name')
            email = request.POST.get('email')
            message = request.POST.get('message')
            return HttpResponseRedirect(reverse('contact_thanks'))
    else:
        form = ContactForm()
    return render(request, 'contact.html', {'form': form})

def contact_thanks(request):
    return render(request, 'contact_thanks.html')
```

What is the security vulnerability in this code and how can it be prevented?

Velg ett alternativ:

- ☐ The security vulnerability in this code is Cross-Site Request Forgery (CSRF), which allows an attacker to trick a user into performing unintended actions on a website. To prevent this vulnerability, the code should use CSRF tokens to ensure that form submissions are coming from legitimate sources.
- ☐ The security vulnerability in this code is Cross-Site Scripting (XSS), which allows an attacker to inject malicious scripts into a web page viewed by other users. To prevent this vulnerability, the code should sanitize user input and encode any output to prevent the execution of malicious scripts.
- ☐ The security vulnerability in this code is SQL injection, which allows an attacker to manipulate the database by sending malicious SQL queries. To prevent this vulnerability, the code should use parameterized queries and input validation to ensure that user input is safe to use in database operations.

Maks poeng: 1

22 Cryptography (1 point)

Which of the following methods is NOT a recommended approach for generating cryptographic keys?

Velg ett alternativ:

- ☐ Employing a software-based secure pseudo-random number generator with unique seeds
- ☐ Reusing a previously generated key for a new encryption task
- ☐ Collecting entropy from user-generated input, such as mouse movements or keyboard strokes.
- ☐ Deriving keys from a passphrase using a key derivation function
- ☐ Using a hardware random number generator

Maks poeng: 1

23 Vulnerabilities (1 point)

Consider the following code snippet:

```
import xml.etree.ElementTree as ET
xml_string = input("Enter some XML: ")
root = ET.fromstring(xml_string)
```

What is the security vulnerability in this code and how can it be prevented?

Velg ett alternativ:

- ☐ The security vulnerability in this code is SQL injection, which allows an attacker to manipulate the database by sending malicious SQL queries. To prevent this vulnerability, the code should use parameterized queries and input validation to ensure that user input is safe to use in database operations.
- ☐ The security vulnerability in this code is XML injection. An attacker could send a malicious XML payload that includes an external entity reference, allowing the attacker to read arbitrary files on the server. To prevent this type of attack, we can disable external entities in the XML parser
- ☐ The security vulnerability in this code is Cross-Site Scripting (XSS), which allows an attacker to inject malicious scripts into a web page viewed by other users. To prevent this vulnerability, the code should sanitize user input and encode any output to prevent the execution of malicious scripts.

Maks poeng: 1

24 Injection attack (1 point)

A web application allows users to search for files on the server by entering a file name into a search form. The application takes the user's input and runs it as a command on the server using the function `system()`. Which of the following inputs would be an example of a successful command injection attack?

Velg ett alternativ:

- ☐ "file.txt && echo 'hello'"
- ☐ "file.txt | grep 'secret'"
- ☐ "file.txt"
- ☐ "file.txt; rm -rf /"

Maks poeng: 1

25 Vulnerabilities (1 point)

Consider the following code snippet:

```
name = fetchNamefromDatabase()
print('Hello, ' + name + '!')
```

What is the security vulnerability in this code and how can it be prevented?

Velg ett alternativ:

- ☐ The security vulnerability in this code is SQL injection, which allows an attacker to manipulate the database by sending malicious SQL queries. To prevent this vulnerability, the code should use parameterized queries and input validation to ensure that user input is safe to use in database operations.
- ☐ The security vulnerability in this code is Cross-Site Scripting (XSS), which allows an attacker to inject malicious scripts into a web page viewed by other users. To prevent this vulnerability, the code should sanitize user input and encode any output to prevent the execution of malicious scripts.
- ☐ The security vulnerability in this code is Cross-Site Request Forgery (CSRF), which allows an attacker to trick a user into performing unintended actions on a website. To prevent this vulnerability, the code should use CSRF tokens to ensure that form submissions are coming from legitimate sources.

Maks poeng: 1

26 Encrypting messages (1 point)

In an asymmetric key system, each user has a pair of keys: a private key and a public key. To send an encrypted message to someone, what must you encrypt the message with?

Velg ett alternativ:

- ☐ Your private key
- ☐ The recipient's private key
- ☐ Your public key
- ☐ The recipient's public key

Maks poeng: 1

27 CVE and CVSS (1 point)

Which of following most accurately describe the purposes of CVE and CVSS?

Velg ett alternativ:

- ☐ CVSS provides a list of top vulnerabilities. CVE is a list of scores for vulnerabilities in a system.
- ☐ CVSS provides a score that indicates the severity of a vulnerability. CVE integrates all the security tools available in an organization and automates incident responses.
- ☐ CVSS provides a score that indicates the severity of a vulnerability. CVE is a list of publicly known vulnerabilities containing ID numbers, descriptions, and references.
- ☐ CVSS is a "low and slow" style of attack executed to infiltrate a network and remain inside undetected. CVE is a list of publicly known vulnerabilities containing ID numbers, descriptions, and references.

Maks poeng: 1

28 CIA (1 point)

Which of the following describes the CIA triad when applied to software security?

Velg ett alternativ:

- ☐ Confidentiality prevents unauthorized access, integrity prevents unauthorized modification, and availability deals with countermeasures to prevent denial of service to authorized users.
- ☐ Confidentiality deals with countermeasures to prevent denial of service to authorized users, integrity prevents unauthorized modification, and availability prevents unauthorized access.
- ☐ Confidentiality prevents unauthorized access, integrity prevents unauthorized modification, and availability deals with countermeasures to prevent unauthorized access.
- ☐ Confidentiality prevents unauthorized modification, integrity prevents unauthorized access, and availability deals with countermeasures to prevent denial of service to authorized users.

Maks poeng: 1

29 Captcha (1 point)

Consider a web application that allows users to create accounts, login, and access sensitive data. Which of the following statements is true about the use of captchas and other security measures in secure coding practices?

Velg ett alternativ:

- ☐ Captchas are an effective security measure that can prevent automated attacks and protect user data, and should be used as the primary means of preventing such attacks.
- ☐ Captchas are effective in preventing automated attacks, but can also be bypassed by sophisticated attackers using machine learning or other advanced techniques. Therefore, they should not be used in secure coding practices.
- ☐ Captchas can be effective in preventing automated attacks, but should not be relied upon as the sole means of protecting user data. Additional security measures, such as input validation, parameterized queries, and user authentication and authorization, should also be used.
- ☐ Captchas are unnecessary and can actually decrease the security of a website by creating a false sense of security, and should not be used in secure coding practices.

Maks poeng: 1

30 Crypto concepts (1 point)

Which statement regarding cryptography concepts is FALSE?

Velg ett alternativ:

- ☐ Symmetric key algorithms are typically faster than asymmetric systems.
- ☐ Symmetric key algorithms are often referred to as public key algorithms.
- ☐ ECC is an example of an asymmetric public key cryptosystem.
- ☐ Symmetric key algorithms use the same private key for encryption and decryption.

Maks poeng: 1

31 Vulnerabilities (1 point)

Consider the following code snippet:

```
#include <stdio.h>
#include <string.h>
int main() {
    char data[5];
    strcpy(data, "Hello World");
    printf("%s\n", data);
    return 0;
}
```

What is the security vulnerability in this code and how can it be prevented?

Velg ett alternativ:

- ☐ The security vulnerability in this code is Cross-Site Scripting (XSS), which allows an attacker to inject malicious scripts into a web page viewed by other users. To prevent this vulnerability, the code should sanitize user input and encode any output to prevent the execution of malicious scripts.
- ☐ The security vulnerability in this code is SQL injection, which allows an attacker to manipulate the database by sending malicious SQL queries. To prevent this vulnerability, the code should use parameterized queries and input validation to ensure that user input is safe to use in database operations.
- ☐ The security vulnerability in this code is buffer overflow, which allows an attacker to overwrite memory beyond the bounds of the buffer and potentially execute arbitrary code or cause a denial of service. To prevent this vulnerability, the code should use safe functions, such as strncpy() and strlcpy(), to copy strings and ensure that the buffer is not overflowed.

Maks poeng: 1

32 Symmetric encryption (1point)

Which of the following is a disadvantage of the symmetric encryption compared to asymmetric encryption?

Velg ett alternativ:

- ☐ Key management
- ☐ Key strength
- ☐ Key size
- ☐ Speed

Maks poeng: 1

33 Threat agents (1 point)

Which of the following statements best describe the characteristics of different threat agents?

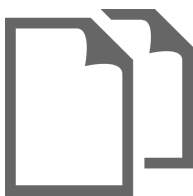
Velg ett alternativ:

- ☐ CEO criminals are highly skilled and motivated to spy on their own employees. Geeks are individuals who perform hacking activities for personal gain or to cause harm. Script Kiddies are typically young or inexperienced individuals who use pre-packaged tools or scripts to launch simple attacks on websites or online services.
- ☐ Swamps are associated with online harassment and bullying. Often very skilled and with many resources. Crooks are motivated by financial gain and may target individuals or organizations to steal sensitive data or extort money. Cyber warriors are independent hackers that attack other systems mainly based on their cultural beliefs.
- ☐ Terrorists have limited skills, but can be highly motivated and have enough resources to finance others to perform cyber attacks. Geeks are individuals who perform hacking activities legally and with the intention of improving security. Hackers-for-hire are individuals or groups who offer their services to conduct cyber attacks on behalf of others.
- ☐ Insiders know the systems well and have access, and therefore do not need many resources to perform attacks against their own organisation. Terrorists use methods such as penetration testing, social engineering, and vulnerability scanning. Spooks are hired by organizations to identify and exploit vulnerabilities in their systems in order to improve security.
- ☐ Geeks are driven by curiosity, have technical skills and unlimited resources. Cyber warriors are malicious individuals or groups who seek to exploit vulnerabilities in systems for personal gain or to cause harm. Insiders may be motivated by financial gain, revenge, or ideology and can be particularly difficult to detect and prevent.

Maks poeng: 1

Question 1

Attached



Case description: Video recordings of football games

A local (Norwegian) football club wants to use video recordings of their matches to analyse their play and stream to people not able to attend the games. The team has bought camera equipment and cloud storage from a UK supplier, and pays a monthly subscription fee for them to manage the video recordings. The club wants to offer these services to teams with players ranging from ages 14-17 (youth teams) and 18+ (adults).



The season is about to start when somebody informs the club that video can be regarded as personal information, and explicit consent needs to be obtained from anyone appearing in the video recordings. The club contacts you to help them out developing an appropriate consent system. The system should also be used to inform spectators (through a privacy notice):

- That video recording is taking place.
- Which areas will be recorded.
- What the video will be used for.
- Who can see it.
- How it is stored and for how long.
- Who to contact for more information.
- How to object.

During the first meeting with the club you come up with the following business goals for a web-based system:

- BG1: Make it easy to give spectators access to the privacy note (e.g., through QR codes).
- BG2: Obtain consent from both home and away teams before recording starts (all players).
- BG3: Know who has been recorded.
- BG4: Keep evidence of the consent.
- BG5: Allow consent to be withdrawn.
- BG6: If the players are below 15 years of age, parents or guardians need to provide the consent.

Part 1 tasks (26 points in total)

In this part you continue with a risk-based assessment based on RMF (Risk Management Framework) of the web-based system described in the case. You can disregard the video recording part of the system, as this is not something you will develop.

If you feel that any of the tasks require information that you do not find in the text, then you should

- Document the necessary assumptions (e.g. technology, standards, software, design choices.)
- Explain why you need them

Your answers should be brief and to the point.

Task 1: Identify the business assets (at least 4) and stakeholders that you find most relevant. (3 points)

Task 2: Identify business risks for the system (at least 5) and link them to the business goals. You may define additional business goals you see fit. (3 points)

Task 3: Describe textually the main elements of a data flow diagram (DFD) representing the system (you can sketch one on paper and then describe the elements you have drawn). You can assume that a Single Sign On solution will be used to simplify authentication. Describe possible attack points in relation to the DFD. (5 points)

Task 4: Based on your threat model (data flow diagram), identify at least 5 technical risks. You should describe necessary assumption related to the technology. Link these to the business risks. (4 points)

Task 5: Select at least 3 technical risks and define one well-formulated security requirement for each as part of a risk mitigation strategy. (3 points)

Task 6: A typical Norwegian football club is run by volunteers, often with limited knowledge about GDPR and technical skills, and usually with a tight budget. Write a short reflection about GDPR challenges you think such organisations tend to face. Who should be responsible in case of a privacy breach? Why is it so important to take GDPR seriously when recording children? (5 points)

Task 7: Write a short reflection where you argue that the club should disregard streaming the games to remote spectators, but rather focus on keeping the video for local analysis only. (3 points)