UNIVERSITETET I BERGEN

KANDIDAT

112

PRØVE

INF226 0 Programvaresikkerhet

Emnekode	INF226
Vurderingsform	Skriftlig eksamen
Starttid	05.12.2025 14:00
Sluttid	05.12.2025 17:00
Sensurfrist	--
PDF opprettet	15.01.2026 10:44

Information

Oppgave	Tittel	Oppgavetype
i	Practical information	Informasjon eller ressurser

Knowledge

Oppgave	Tittel	Oppgavetype
1.1	Authentication and Authorization	Flervalg
1.2	Containerisation	Flervalg
1.3	CORS and CSP	Flervalg
1.4	Buffer Overflows	Flervalg
1.5	Coding practices	Flervalg (flere svar)

Skills and General Competence

Oppgave	Tittel	Oppgavetype
2.1	Expera	Langsvar
2.2	Cookies	Flervalg (flere svar)
2.3	Content Security Policy	Fyll inn tekst
2.4	DREAD	Langsvar
2.5	Web Architecture	Langsvar
2.6	Code Injection	Langsvar

Portfolio

Oppgave	Tittel	Oppgavetype
3.1	Assignments	Muntlig

3.2 Injected Holidays □ Langsvar

□

1.1 Authentication and Authorization

Each of the following six questions awards 1 point for a correct answer, deducts 0.5 points for an incorrect one, and gives 0 points for no answer.

1. Which of the following best describes *authentication*?

Select one alternative

- ○ Encrypting data before storage
- ● Verifying the identity of a user or system
- ○ Validating that a user has permission to perform a specific action
- ○ Ensuring transmitted data has not been altered

2. Which of the following is an example of *multi-factor authentication (MFA)*?

Select one alternative

- ○ Using single sign-on (SSO) through a corporate account
- ○ Entering only a username and password □
- ○ Scanning a QR code to log in
- ● Logging in with a password plus a one-time code sent to a phone

3. What is the primary purpose of *OAuth 2.0* when used in *authentication* workflows?

Select one alternative

- ● To allow users to authenticate using a third-party provider without sharing passwords
- ○ To provide end-to-end encryption between clients
- ○ To manage user roles and permissions inside an application
- ○ To protect APIs through rate limiting

4. Which of the following best describes *authorization*?

Select one alternative

○ Verifying the identity of a user

○ Detecting malicious activity on a system

○ Ensuring that data remains confidential during transmission

● Determining what an authenticated user is allowed to do

5. In role-based access control (RBAC), which of the following is the most accurate statement?

Select one alternative

● Users receive permissions by being assigned to roles

○ Roles are automatically generated based on user activity

○ Roles only apply to file systems, not applications

○ RBAC eliminates the need for authentication

6. Which of the following distinguishes capability-based access control from access-control lists (ACLs)?

Select one alternative

○ Capabilities restrict permission checks to operating-system kernels only

○ Capabilities cannot be delegated from one process to another

○ Capabilities enforce authorization through roles instead of groups

● Capabilities are unforgeable tokens that grant access directly to objects

Maks poeng: 6

1.2 Containerisation

Each of the following three questions awards 1 point for a correct answer, deducts 0.5 points for an incorrect one, and gives 0 points for no answer.

1. What is the primary purpose of containerization (e.g., Docker) in software systems?

Select one alternative

● To isolate processes and control their access to system resources

○ To replace virtual machines entirely

○ To encrypt all application data at rest

○ To run applications across multiple physical servers simultaneously

2. What best describes the filesystem behavior of a container?

Select one alternative

● Containers use an isolated root filesystem that can map selected host paths

○ Each container has an entirely separate physical disk

○ Containers must run from a single global read-only image

○ Containers share the host filesystem without restrictions

3. Which of the following is a key security benefit of using containers?

Select one alternative

○ Containers prevent any type of network communication

○ Containers eliminate the need for operating-system security patches

● Containers limit what system resources an application can access

○ Containers guarantee that no process can ever access host hardware

Maks poeng: 3

1.3 CORS and CSP

Each of the following three questions awards 1 point for a correct answer, deducts 0.5 points for an incorrect one, and gives 0 points for no answer.

1. What is the primary purpose of CORS (Cross-Origin Resource Sharing)?

Select one alternative

○ To control which cross-origin requests the browser may treat as safe to reveal to the calling page

○ To ensure that cross-origin requests are encrypted and protected from man-in-the-middle attacks

● To prevent websites from loading untrusted scripts

○ To block all cross-origin requests entirely

2. What security risk is CSP (Content Security Policy) primarily designed to mitigate?

Select one alternative

○ DNS spoofing

○ SQL Injection

● Code Injection / Cross-Site Scripting (XSS)

○ Cross-Site Request Forgery (CSRF)

3. Which statement best describes the relationship between CORS and CSP?

Select one alternative

○ Both mechanisms block cross-origin HTTP requests from being sent

○ CSP replaces the need for CORS in modern browsers

● CORS controls which origins may access a server’s resources, while CSP restricts what external content a page may load and execute

○ CORS prevents injection attacks, while CSP prevents cross-origin data theft

Maks poeng: 3

1.4 Buffer Overflows

Each of the following three questions awards 1 point for a correct answer, deducts 0.5 points for an incorrect one, and gives 0 points for no answer.

1. What is the primary risk of a classic stack buffer overflow?

Select one alternative

○ Unauthorized modification of the filesystem

○ Preventing the program from being compiled

○ Causing the CPU to overheat

● Overwriting adjacent memory, potentially altering the return address

2. What is the main purpose of a stack canary?

Select one alternative

○ To encrypt stack memory

○ To increase stack size dynamically

○ To randomize memory addresses

● To detect stack corruption before a function returns

3. Which of the following best describes Address Space Layout Randomization (ASLR)?

Select one alternative

○ A method to detect memory corruption at runtime

● Randomizing the memory locations of code and data to make exploitation harder

○ Preventing buffer overflows by limiting stack size

○ Encrypting pointers stored on the stack

4. Which statement about stack overflow mitigation techniques is true?

Select one alternative

○ Stack canaries increase stack size to prevent overflows

○ ASLR requires modifying the application code to work

○ ASLR alone can completely prevent all stack overflows

● Stack canaries and ASLR are complementary, both reducing the likelihood of successful exploits

---

Maks poeng: 4

□

1.5 Coding practices

(You get fraction of a point for each correct choice, a small penalty for each incorrect choice, and 4 points in total if everything is correct.)

Which of the following are benefits of using a strong static type system in a programming language?

Select one or more alternatives

- ☑ Making code more self-documenting
- ☑ Reducing the likelihood of type-related security vulnerabilities
- □ Eliminating the need for input validation
- □ Preventing all runtime exceptions
- ☑ Catching certain classes of errors at compile time

Which of the following are recommended practices for handling exceptions securely?

Select one or more alternatives

- ☑ Catching exceptions broadly, so the program doesn't crash
- □ Sanitizing error messages before returning them to users
- ☑ Logging exceptions in a safe, non-sensitive way
- □ Displaying raw stack traces for easy debugging

Which coding practices support writing more secure software?

Select one or more alternatives

- ☑ Using immutable data structures when appropriate
- □ Disabling compiler warnings to reduce confusing noise
- ☑ Applying the principle of least privilege in code and configuration
- ☑ Minimizing the use of global mutable state
- ☑ Documenting assumptions and using type annotations

Maks poeng: 4

2.1 Expera

You have been hired as a security consultant for the University's new digital exam solution, *Expera*. It will be similar in functionality to *Inspera Assessment* (which you're currently using), but with improved security.

Expera will be an integrated system include tools to create and grade exams (for teachers/lecturers), to manage exams (for administrators) and to take/answer exams (for students).

Consider the system in terms of *Confidentiality, Integrity, Availability and Traceability* (CIA-T) and the *STRIDE* threat model. What could go wrong? What would the consequences be? What are the main threats and risk factors? **Create a simple threat model for the Expera exam system.** *(2–4 paragraphs)*

Write your answer in the box below. Changes are saved automatically.

What could go wrong?

In terms of confidentiality, the database could leak with the answers to students. With integrity, someone could impersonate another student. A student could get elevated privileges. With availability, if we use a lot of third party services they could go down, ex Cloudflare this morning (12.5.2025). With Traceability we could have multiple graders grading and one of them grading unfairly. It is also a scenario where if we are not able to trace who deliver witch exams

What would the consequences be?

database leak: If the student-ids are mapped to the candidate numbers with plain text, the confidential part of the exam would be compromised. The person grading the exams could get hold of which students answered which exam and the entire grading process would also be compromised.

If a student manages to impersonate another student during the exam (exam staff often doesn't pay much attention) the results would loose its integrity. If a student gets elevated privileges, they could alter their and others grades, loosing integrity.

If a third party service we depend on goes down, during, before or after the exam this will affect students who cannot deliver their exam, preparation for students, and grading times for lecturers.

If we are not able to trace who graded which exams, we would not be able to track unfair grading, and if they complain, we would not be able with certainty to give them a new grader. And if we do not know who answered which exam we would not be able to grade at all.

What are the main threats and risk factors?

Spoofing, someone trying to impersonate a lecturer and get privileges to grade exams (elevated privileges). This would also be tampering if they got this access and started changing grades. Another threat is that students are able to write freely into answer boxes, here someone could try to enter malicious code. A student could also try and say that the delivered answers are not theirs (repudiation). There is also a big risk with third party services going down and our system not being able when needed. Another Dos risk if students try to DDos attack the site during delivery of the exam, and the proper answers doesn't get registered. Students having malware on their computer to try and compromise / cheat during the exam.

Ord: 391

Maks poeng: 10

2.2 Cookies

Your first task on the *Expera* project is to configure login and session management. It works as follows:

- Students will log in by going directly to expera.no.
- After logging in, their session is tracked with a cookie.

Note that:

- The students may also have access to other sites that you don't control.
- The company has several products, and different subdomains of expera.no might be used for unrelated services.
- You should assume that there may be flaws in the rest of the system.

How would configure the session cookie?

Select one or more cookie attributes:

☑ HttpOnly

☑ Secure

☑ SameSite=Strict

☑ Domain=expera.no

☐ SameSite=None

☐ SameSite=Lax

(1 point for each correct choice, -0.5 points for an incorrect one, and 4 points in total for everything correct.)

Maks poeng: 4

2.3 Content Security Policy

You also need to configure a Content Security Policy for Expera. You know that your competitor, Inspera, has had problems with JavaScript code injections, so you want to be sure that doesn't happen to you. Fill in the appropriate CSP directives:

- All scripts need to be under your control.
- Styles should only be loaded if they match a 256-bit hash.
- Images can also be loaded from uib.no

Content-Security-Policy: default-src 'self' 'unsafe-inline';

script-src 'self' ;

style-src '$HASH' ;

img-src 'self' 'uib.no' ;

(use $HASH or $NONCE as a stand-in for hash or nonce values, if needed)

(1 point for each correct answer, no penalty for incorrect answers.)

Maks poeng: 3

□

2.4 DREAD

Before you deploy your security improvements from 2.2 and 2.3, you discover that certain malicious sites are able to change a student's exam answers if visited by the student during the exam. Also, if a student is logged in to the exam system outside of the exam (e.g., to check their grades), information about the student can leak to malicious sites the student are visiting. With your improvements deployed, the problem goes away.

***a)* [10%]**

Analyse and rate this threat in terms of *DREAD* *(Damage, Reproducibility, Exploitability, Affected users, Discoverability)*. You can rate each category as *low, medium,* or *high* risk. **Explain your thinking.** *(2–4 paragraphs)*

***b)* [5%]**

Under the GPDR, what responsibilities would Expera have if this actually happened while students were using the system? **Explain.** *(1–2 paragraphs)*

***c)* [5%]**

Which of the security mechanisms from the previous tasks (2.2–2.3) do you think would mitigate this vulnerability? **Explain.** *(1–2 paragraphs)*

Write your answer in the box below. Changes are saved automatically.

a)
I assume the malicious site/sites are NOT trusted, common websites like lovdata.no and is not something that you open regularly during exams.

Damage - High since it would ruin the exam/semester for the student.

Reproducibility - low since it requires a specific combination of sites to be open simultaneously, and the pages would need to most likely be exam relevant so that students open during / after the exam. Not easy to create webpages that students would visit.

Exportability - High risk since it is easy to create the CSRF attack / malicious site. The challenge is to make students visit you site while also being logged into the exam system.

Affected users - low since it is probably not many students simultaneously visiting the exam site + the malicious site during the exam.

Discoverability - low since it would get discovered really quickly when students didn't get the grades they expected, and then saw that the answers was changed. They would report and complain about the exam.

b)
According to GPDR Expera has 72 hours to notify the authorities. Since this is high impact they also have to notify the affected parities. They will have to document the incident and

provide the necessary fixes.

c)

All of the CSRF prevention mechnisms. HttpOnly, and SameSite=strict are the two which would help the most. HttpOnly disables js so that js cannot reach the cookie. and sameSite=strict will make it so the cookie is only sendt when the request is from the same origin.

Ord: 253

Maks poeng: 20

2.5 Web Architecture

On the backend, web applications are often placed behind a *reverse proxy* webserver like Nginx. **What are the benefits of this, compared to letting the web application serve clients directly?** *(~1 paragraph)*

Write your answer in the box below. Changes are saved automatically.

We can use reverse proxies as a virtual patching tool. This add a layer of security allowing us to easily deploy and try patches during an attack.

Ord: 27

Maks poeng: 6

2.6 Code Injection

Assume that you're working together with developers who haven't taken INF226 or a similar software security course. Your team is doing “full stack“ web development, so you deal with both frontend and backend code.

Explain to your fellow developers what they should do / what practices they should follow to avoid making code injection vulnerabilities. *(2–3 paragraphs)*

You can assume that your team works with JavaScript, Python or Java, SQL, HTML and CSS, if that helps.

Write your answer in the box below. Changes are saved automatically.

Firstly we should mostly never let user supplied data contain unescaped HTML tags on our output. The best way to do this is to use a trusted library for escaping user supplied data. For tags it is important to escape tags such as: < > & " ', and change them out with something that istead just renders it as plaintext. Another measure agains XSS is to implement a CSP and have the HttpOnly flag if possible.

We also need to watch out for SQL injections thru our code. We should always first never only do client-side checks but always do our main checks in the backend. Most important is to always use prepared statements and never string concatenation.

When we do our input checks is the backend we should always use negative space programming to catch errors as fast as possible.

Ord: 142

Maks poeng: 7

3.1 Assignments

The results from the portfolio assignments will be inserted here.

Maks poeng: 30

3.2 Injected Holidays □

This is not a task and you won't get any points for it, but you can write something nice if you like, or use it as a scratchpad – and if Inspera still allows code injections, you'll see a bouncing Christmas tree!

Merry Xmas and a Happy New Year!

a

Ord: 1

Maks poeng: 0

□