UNIVERSITETET I BERGEN

KANDIDAT

258

PRØVE

INF226 0 Programvaresikkerhet

Emnekode	INF226
Vurderingsform	Skriftlig eksamen
Starttid	22.11.2022 14:00
Sluttid	22.11.2022 17:00
Sensurfrist	--
PDF opprettet	03.09.2023 11:39

[Practicalities]

Oppgave	Tittel	Oppgavetype
ℹ	Practical information about the exam	Informasjon eller ressurser

[Exam Questions]

Oppgave	Tittel	Oppgavetype
1	Mixed Questions	Flervalg
2	Web Security	Tekstfelt
3	Software Design	Langsvar
4	Type Safety	Langsvar
5	Web App	Langsvar
6	Verification Issues	Langsvar

[Assignment Results]

Oppgave	Tittel	Oppgavetype
7	Compulsory Assignment Results	Muntlig

1 Mixed Questions

(1 point for each correct answer, 0.5 points for partially correct, 0 for no answer, -1 point for each wrong answer. Min 0 points, max 7 points.)

Authentication is ideally done by…
Select one alternative:

- proving you can perform some particular action
- either an elliptic-curve digital signature (ECDSA) with a key size of at least 160 bits, or traditional DSA with a key size of at least 2048 bits
- showing two or more of: something you know, something you are and something you have
- showing that you know a shared secret

A *stack frame…*
Select one alternative

- is the context of operations that happens between a push and pop of the same element
- should be placed at a random memory location to prevent attacks
- stores data and metadata for a function call
- 'frames' the data on the stack with special guard values to prevent tampering

To prevent SQL injection attacks, you should…
Select one alternative

- use prepared statements
- use stored procedures
- replace every single quote with a double quote in user input
- always use a NoSQL database instead where possible

The phrase *'escaping characters'* means…

Select one alternative

- recoding input strings in a more appropriate character set
- the characters that are written past the end of the buffer in a buffer overflow
- replacing syntactically significant characters to prevent misinterpretation
- throwing an exception on illegal characters in the input

The set of all possible points in a system where an unauthorized users can enter or access/change data is…

Select one alternative

- the attack set
- the attack point
- the attack surface
- the attack vector

The three fundamental aspects (or properties, criteria, etc.) of information security are usually considered to be…

Select one alternative

- maintainability, reliability and security
- encryption, hashing and public key infrastructure
- risk, vulnerability and mitigation
- confidentiality, integrity and availability

With the OAuth 2.0 open standard a user can…

Select one alternative

- use a third-party service (such as Google, Facebook or FEIDE) to authenticate
- authorize a service to access a third party service on their behalf
- both of these
- none of these

Maks poeng: 7

2 Web Security

Answer the questions briefly (i.e., one or two sentences each).

a) How does a CSRF token help protect against *cross-site request forgery* attacks?

Fill in your answer here

The CSRF token helps protect against corss-site request forgery attack by generating a seassion token which is completely random and autenticates that the current user that is logged in is the actuall user that is operating the webiste.

b) Why is a digital certificate required for websites that are accessed with *https* (i.e., HTTPS over TLS)?

Fill in your answer here

A digital certificate is required beacuse we need to verify that this perticular site says who they are. Whitout this certificate we would not be able to prove that the

c) Why do you think some web APIs – like access to camera, microphone, sensors or notifications – are restricted to only work in *secure contexts* (i.e., on webpages served over HTTPS)?

Fill in your answer here

Web APIs that is described above is important to send over HTTPS becuase without the information is sent unencrypted over the internett, and anybody could just listen to the internett trafic and gather information from whom that uses the APIs

d) How could a hacker attack have legal implications for a company?

Fill in your answer here

If a hacker attack is successfull, and is asble to compromise private data on users of that company. The company might face leagal charges, and fines for their lack of security messures on the stored data.

Maks poeng: 12

3 Software Design

Discuss the following questions briefly (e.g., 2–5 sentences / a paragraph or so).

a) *Encapsulation* means bundling data with the methods that operate on the data (i.e., as *objects*), and preventing other parts of a program from accessing the data directly (e.g., with *private* fields in Java). How can this help prevent bugs and security problems? **Give three examples** (a couple of sentences each).

b) *Immutable* data structures or objects can't be modified after they're created. **Explain briefly** how this might be beneficial for software security.

c) *Abstraction* is crucial when building software systems – hiding unnecessary implementation details behind modules / interfaces / classes / methods, and reusing existing libraries. Do you see any security implications of this? (E.g., would a software design heavily based on abstraction be beneficial and/or detrimental to security?) **Explain.**

Fill in your answer here

a)
With the private field in Java we can encapsulate a variable or function to be just run by that specific class, this is important because if we would set everything as public another program could come and change that variable when it was not supposed to. Some examples of this may be the value of a bank account, grades on studentweb, and administration access to a computer system.

b) Immutable data structures is beneficially for computer structures because this means when the object is set in cannot be tampered with. Things that might happen if we didn't have immutable object is that they might get accidentally overwritten when we didn't attend to. This also means that we have some sort of anker point in our program.

c) Sometimes when implementing something you don't need to know everything about an object to define or represent that object, this is where abstraction comes into play. As an example when implementing a car object, you don't need to know how many screws are in the car, or if the car has cup-holders or not. All this unnecessary information can be ignored and only information that is important for the programs function should be implemented for the program to do it's job.

Ord: 209

Maks poeng: 12

4 Type Safety

A *type safe* (sometimes referred to as *strictly typed*) programming language has rules for how data (variables or values) can be used (i.e., what operations or methods can be used on a value).

a) How might this be beneficial for software security? Give two examples of security related bugs/issues that can be prevented by the type system.

b) Some languages are *dynamically typed* (checks types at runtime – Python, for example) while others are (mostly) *statically typed* (checks types at compile time – Java, for example). Does this make a difference from a security point of view?

c) In the first compulsory assignment you performed buffer overflow attacks against C programs. Would you consider C to be a type safe language?

d) Would a type safe programming language help protect against malicious user input?

(Write 1–5 sentences for each question.)

Fill in your answer here

a) When talking about software security strictly typed languages can be beneficial in the instance where a function only should take in an integer, but instead the variable contains any other object the system might crash which can compromise security. Another bug that can occur when not using a strictly typed programming language is the program might not do as intended. For example with indentation languages such as python your indentation might be wrong and it might not be as easy to spot the indentation bug.

b) If would argue yes. If there is a small bug in the code that is almost never run, the program might be put out in production even if there exist a bug in the system. And it is not until the edge case is met (potentially by an attacker) that the bug is spotted.

c) I would not consider C to be a type safe language. C is a language that is closer to the computer that most other languages, here you have a lot of possibilities to refactor your own code, and make it exactly as you wish. But on the other hand this requires a lot more attention to whether or not the application is safe. So for example in the assignment 1 we can sett the buffer to be 8 bytes but also set the input to well over 8 bytes which makes buffer overflow attacks possible in the program.

d) No a type safe programming language would not help against malicious user input. This can be proven by referring to injection attacks, it's not always easy for program to differentiate between whether the input of the user is supposed to be data or code.

Ord: 285

Maks poeng: 12

5 Web App

(Set aside some time to read and understand the context – the expected answers are short, but you'll need some time to figure things out.)

You've been hired for the summer to work on Anya's pet project. She has a web site for programming courses where students can do exercises, get help from TAs, look at their progress and grades so far. Each student logs into their own account on the site, and login sessions are tracked with cookies. Anya asks you to add some new features:

a) First of all, she wants to make it easy to make code examples to illustrate program flow, objects in memory, stack layout and such – like a friendly, graphical debugger. For convenience and flexibility, she wants the programs to be specified as parameters rather than having to upload the example programs to the site. Query parameters can be included in a URL, so this means you (or anyone) could just make a (really long) link, and post it in an announcement or chat or somewhere, and students would end up in the debugging view with the program preloaded when they click on the link. She suggests that the URLs have this form:

https://example.com/c00l_stuff?program=…&commands=…

Where the *program* parameter contains the source code and the optional *commands* parameter is used to set up the debugger (like *gdb* commands, *run*, *step*, *frame*, etc). She says this will be safe, because the connection is encrypted, the code will only be running in the browser anyway, and the system is only used at our University. You're skeptical and think such links could be exploited to harm or play pranks on other students.

What are the main risks in such a scheme? Would you recommend any restrictions to make it safer, or do you see an alternative way to achieve the same goal? **Discuss**. *(A few sentences or a paragraph.)*

b) Anya has already started implementing a prototype. This is her JavaScript code, to be run in the browser when the page is loading:

```
const display = document.getElementById('display'); // HTML element where program will be displayed
if (window.location.search) { // do we have URL parameters?
    // decode parameters
    const usp = new URLSearchParams(window.location.search);
    // display the program text
    const program = usp.get('program');
    display.innerHTML = '<pre class="javaCode">' + program + '</pre>';
    // run the commands
    const shellCmds = usp.getAll('commands');
    shellCmds.forEach((cmd) => {
        shell.runCommand(cmd);
    });
}
```

You immediately notice a classic vulnerability in the code. **What kind of vulnerability do you see? How would you fix it?** *(1–3 sentences.)*

c) Anya wants students to be able to save their code to GitHub. Her favourite meme app can post to Facebook, so she's pretty confident this should work. **How would you** go about letting the site

interact with GitHub's API on the user's behalf? *(One sentence is enough.)*

d) You implement the code to access GitHub, but you get this strange error message when you try to use it? **What's going on? Why would the browser stop requests to other sites?** *(1–3 sentences.)*

```
» req = await fetch('https://api.github.com/users/anyahelene/repos', {method:'GET', headers: {authentication: 'Bearer: ' +
  token, 'content-type':'application/json'}})
```

```
Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at https://api.github.com/users
/anyahelene/repos. (Reason: header 'authentication' is not allowed according to header 'Access-Control-Allow-Headers' from
CORS preflight response). [Learn More]
Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at https://api.github.com/users
/anyahelene/repos. (Reason: CORS request did not succeed). Status code: (null). [Learn More]
Uncaught (in promise) TypeError: NetworkError when attempting to fetch resource.
```

Fill in your answer here

a) Since the code is run on a browser links like this could potentially be used to Cross-Site Scripting and Cross-Site request forgery where this code could attain information that is stored in the browser such as cookies or be used to attain information from other webpages that you have logged into. So if the other websites you have used have bad security features on their website this link could potentially send information that was not supposed to be obtained by the attacker.

b) Some vulnerability that comes to mind when looking over this code is that there is some string concatenation in this code which could be used by the attacker to inject malicious code. Since it is stored as a string the program cannot always differentiate what is supposed to be code, and what is supposed to be data. If the attacker inject some form of code as it's data, the program could be confused and run some data as code.

c) We cannot always trust that a site is acting as it should, there could be some forgery happening with your site and therefore the site should not interact with Github's API on the users behalf.

d) It seems like when we try to access this repository we do not have the right access control rights to be able to send a GET request to the site we want to access. Another probable cause is that the Same site cookie is set to strict such that we cannot send a GET request to the site we want to access. This is because we get a Cross-origin Resource Sharing Blocking, and this is again used to prevent misuse of data and various cross-site attacks, which means the browser will prevent loading certain resources or making calls to sites that don't share the same origin.

Ord: 306

Text version of the code in the image:

```
const display = document.getElementById('display'); // HTML element where program will be
displayed
if (window.location.search) { // do we have URL parameters?
    // decode parameters
    const usp = new URLSearchParams(window.location.search);
    // display the program text
    const program = usp.get('program');
    display.innerHTML = '<pre class="javaCode">' + program + '</pre>';
    // run the commands
```

```
    const shellCmds = usp.getAll('commands');
    shellCmds.forEach((cmd) => {
        shell.runCommand(cmd);
    });
}
```

Text version of the error message:

>>> req = await fetch('https://api.github.com/users/anyahelene/repos', {method:'GET', headers: {authentication: 'Bearer: ' + token, 'content-type':'application/json'}})

Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at https://api.github.com/users/anyahelene/repos. (Reason: header ‘authentication’ is not allowed according to header ‘Access-Control-Allow-Headers’ from CORS preflight response).

Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at https://api.github.com/users/anyahelene/repos. (Reason: CORS request did not succeed). Status code: (null).

Uncaught (in promise) TypeError: NetworkError when attempting to fetch resource.

Maks poeng: 12

6 Verification Issues

Earlier this month, Twitter changed their “blue checkmark” account feature from being used for *verified accounts* (account that were *authentic, notable* and *active* – representing a verified actual person or company) to being a subscription feature linked to the $8/month *Twitter Blue* subscription feature. The internet promptly exploded with humorous and malicious fake ‘verified’ accounts – in one instance, a pharmaceutical company lost billions in market capitalization after a fake account announced they would provide free insulin *(Fig. A – below the text area)*.

For an example of the sort of problem verified accounts were meant to prevent, consider the tweet in *Fig. B* below and the two account profiles in *Fig. C* and *D* (all are actual screenshots of existing accounts and tweets, not manipulated images.)

Thinking in terms of security properties like CIA(-T)[1], the STRIDE[2] threat model and/or DREAD[3] risk assessment model – what are (some of) the security implications of such a change to Twitter's verification system? Can you think of ways to mitigate the problems? **Explain briefly** (a paragraph / ~3–5 sentences).

([1] Confidentiality, Integrity, Accessibility, (Traceability); [2] Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privileges; [3] Damage, Reproducibility, Exploitability, Affected users, Discoverability)

Fill in your answer here

I would say that twitter doesn't directly violate the property of CIA-T and some of the others, because even thought the attacker poses as somebody else they haven't directly used another users account, they have just created an identical account. You could potentially say this is some type of spoofing because they are posing as somebody else, but this includes the tricking of the computer system, which it is not. This type of spoofing or more like phishing attack sets the users of the platform at risk, which with the new twitter blue feature can pose as authenticated and a reliable source for other users at the platform. Personally this was a bad move by twitter, on the base that users of the platform no longer can verify with a quick look that the information that is given by them come from a credible source. Furthermore to mitigate this confusion wether a user is credible or not, I have no clear solution. I would personally just go back to the old system, where twitter needs to verify each and every user by themselves.

Ord: 183

Fig. A:

Fig B:

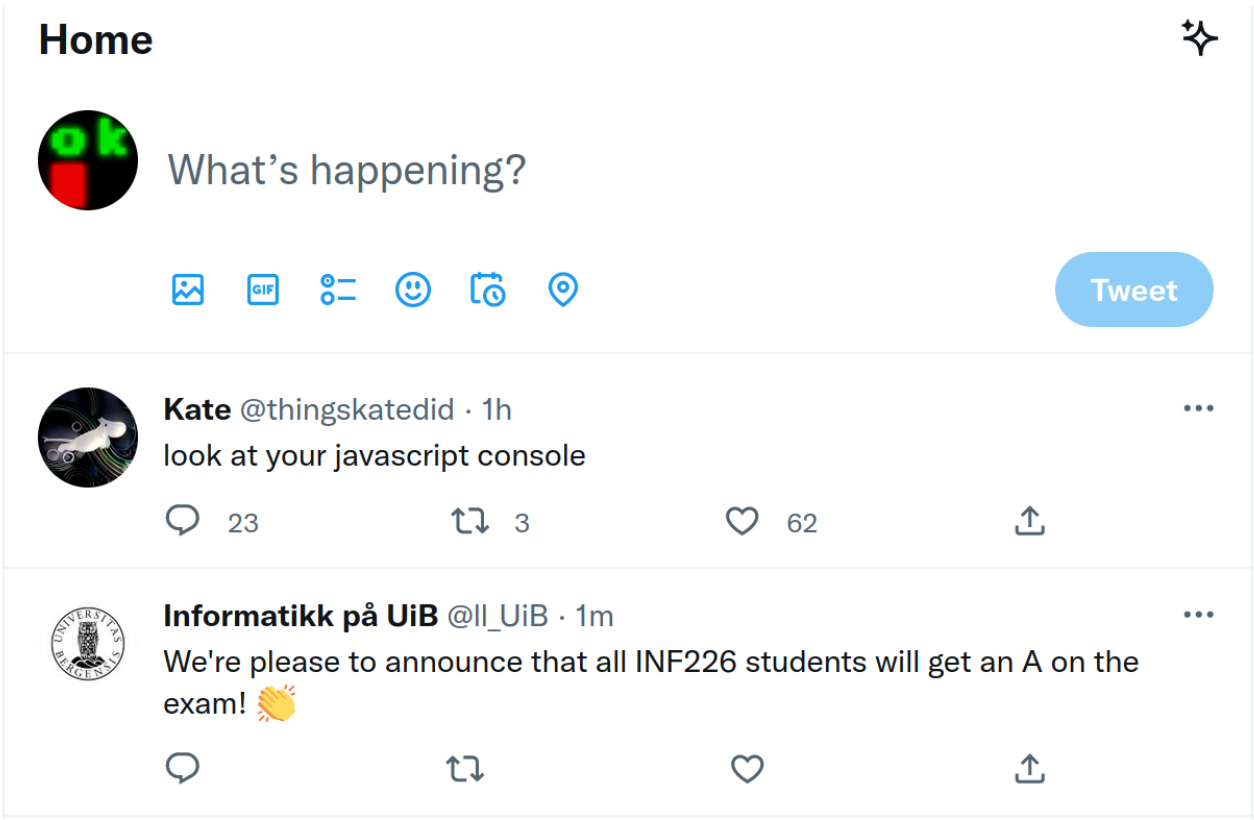

Fig. C: *https://twitter.com/ll_UiB*

Fig. D: *https://twitter.com/II_UiB*

Maks poeng: 5

7 Compulsory Assignment Results

Yay, you're done! *(This ‘exercise‘ is just a placeholder and will be filled in during grading.)*

Maks poeng: 40