English

Universitetet i Bergen
Det matematisk-naturvitenskapelige fakultet

Exam in	:	INF226 Software Security
Semester	:	Spring 2022
Time	:	09:00 – 12:00, 2nd of March 2022
Number of pages	:	5
Permitted aids	:	none

- This exam counts for 60% of your final grade in this course.
- Points on the exercises indicate *approximate* percentage weight on the total final grade.
- Give justifications for your answers, unless otherwise specified.
- Use precise language and make sure to read the exercises carefully.

Exercise 1 (12 points)

Answer the questions below with one or two sentences on each.

a) Where is the stack canary located?
b) What is shell code in the context of buffer overflow exploits?
c) What kind of attacks do prepared statements protect against?
d) Why should un-trusted data not be inserted directly into the HTML code of a web-page?
e) How are the requirements of a key derivation function different from the requirements of a cryptographic hash function?
f) What does the HttpOnly flag signify?

Exercice 2 (10 points)

User authentication is a common challenge with web applications. In this exercise we will discuss some aspects of password based authentication of users.

a) *Server-side password storage*: Explain how to securely store user passwords on the server side, to prevent an attacker from knowing the passwords if the database is leaked. Be specific about which algorithms you would use, and what kind of attacks are prevented by these methods.

b) *Session cookies*: A session cookie is a randomly generated token used to authenticate the user after they have logged in with their password. Explain how access to session cookies is restricted by policy in the web browser, and how a cross-site request forgery circumvents this protection.

Exercice 3 (12 points)

A small health care clinic uses a system with software controlled access cards for security. Each employee has an access card which they use to access the buildings, equipment, medicine lockers and rooms.

The access cards software is currently based on access control lists (ACL). But maintaining the ACL is very tedious, and various security related instances have occured.

For instance, when Benny started as a nurse at the clinic, they forgot to give him access to the medical equipment locker. So, the first morning he was supposed to prepare for an operation he could not get to the equipment and the operation was delayed. Also, Martine recently started working as a secretary, replacing a previous secretary who was, by accident, given access to the medicine lockers and had helped themselves to its content.

After Hanne, the attending physician at the clinic brought this up, the maintainers of the software wants to switch to another access control model.

a) What are the general strengths and weaknesses of ACLs as an access control model? Which of these general strengths and weaknesses apply in the situation above?

b) Suggest a different access control model for the access cards, and give reasons for your choice.

c) Adapt the access control list to your new model. Specify any relevant schemas and fill in the tables based on the data below.

The current access control list is on the next page.

Person	Permission
Anna	Enter main entrance
Anna	Medical equipment locker
Anna	Enter operation hall
Benny	Enter main entrance
Benny	Medical equipment locker
Benny	Enter operation hall
Hanne	Enter main entrance
Hanne	Medical equipment locker
Hanne	Enter operation hall
Hanne	Medicine locker
Ike	Enter main entrance
Ike	Medical equipment locker
Ike	Enter operation hall
Ike	Medicine locker
Linnea	Enter main entrance
Linnea	Medical equipment locker
Linnea	Enter operation hall
Luis	Enter main entrance
Luis	Enter operation hall
Luis	Enter office
Luis	Cleaning cupboard
Martine	Enter main entrance
Martine	Enter office
Martine	Document locker
Sasha	Enter main entrance
Sasha	Enter office
Sasha	Document locker
Vera	Enter main entrance
Vera	Enter operation hall
Vera	Enter office
Vera	Cleaning cupboard

Exercice 4 (12 points)

Below is an excerpt from the InChat source code, after a student has attepted to fix the security holes which were in this function.

```
private void printEvent(PrintWriter out,
                        Channel channel,
                        Stored<Session> session,
                        Message message) {
    out.println("<div class=\"entry\">");
    out.println("    <div class=\"user\">"
             + Encode.forHtml(message.sender) + "</div>");
    out.println("    <div class=\"text\">"
             + Encode.forHtml(message.message));
    out.println("    </div>");
    out.println("    <div class=\"messagecontrols\">");
    out.println("        <form style=\"grid-area: delete;\" action=\"/channel/"
             + Encode.forHtml(channel.name) + "\" method=\"POST\">");
    out.println("        <input type=\"hidden\" name=\"message\" value=\""
             + message.identity + "\">");
    out.println("  "
             + csrftoken(session));
    out.println("        <input type=\"submit\" name=\"deletemessage\" value=\"Delete\">");
    out.print  ("        </form><form style=\"grid-area: edit;\"");
    out.println("action=\"/editMessage\" method=\"POST\">");
    out.println("        ");
    out.println("        <input type=\"hidden\" name=\"message\" value=\""
             + message.identity + "\">");
    out.println("        <input type=\"hidden\" name=\"channelname\" value=\""
             + channel.name + "\">");
    out.println("        <input type=\"hidden\" name=\"originalcontent\" value=\""
             + Encode.forHtmlContent(message.message) + "\">");
    out.println("        <input type=\"submit\" name=\"editmessage\" value=\"Edit\">");
    out.println("        </form>");
    out.println("    </div>");
    out.println("</div>");
}
```

a) Explain why the above code might still be vulnerable to a cross-site scripting attack.

b) Assume that HttpOnly flag is enabled for the session cookies in InChat. How does that affect how an attacker could exploit this vulnerability? Can the attacker still exploit the vulnerability in some way?

c) What additional measures against cross-site request forgery would you recommend for InChat to prevent similar mistakes from being exploitable in the future?

Exercice 5 (14 points)

Imagine an attack scenario where an attacker has gained control over a process on a machine, and can execute code remotely through this process. The attackers next step is escalate their privileges and access other resources on the same system such as:

- the file system,
- other processes and their memory, and
- access the local area network.

In this exercise you can assume the process is running on Linux or another operating system you are familiar with.

a) Which operating system (OS) security mechanisms protect each of these resources from the attacker controlled process?
b) What protective measures can a programmer put in place on a process in order to prevent privilege escalation if the process is taken over by an attacker?
c) On Android, Google's successful mobile operating system derived from Linux, each app get its own UID. Why do you think they made that descision?