NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Computer and Information Sciences

Examination paper for TDT 4237 Software Security

Academic contact during examination: Carl-Fredrik Sørensen

Phone: 951 19 690

Examination date: December 9th, 2015

Examination time: 09:00 AM to 1:00 PM

Permitted examination support material: Code D – No printed or hand-written materials allowed. Approved calculator allowed.

Other information: Exam developed by Carl-Fredrik Sørensen and checked by Per Håkon Meland

Language: English

Number of pages: 5

Number of pages enclosed: 0

Checked by:

____

Date Signature

Introduction

In this exam, you can score a maximum of 70 points. The remaining 30 points for the semester comes from the compulsory exercises.

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

- Document the necessary assumptions
- Explain why you need them

Your answers should be brief and to the point.

Problem 1 – (40 points)

a) (10%) It is hard to keep a private key to a system protected. Describe different methods an attacker may use to get access to such a key, enabling the attacker to get access to valuable digital assets within a company. Note: Do not dig into technical details; give a few examples of each identified method.
b) (5%) Explain the differences between multilevel and multilateral security policies. Give examples of established models of security policies.
c) (5%) When doing a risk assessment of a system, which methods or ways can be used to calculate/place a probability and impact/consequence to rank a risk? Give example(s) to illustrate such a method.
d) (10%) What is the main differences between BSIMM and OpenSAMM? How can a company enhance software security practices by using these frameworks?
e) (5%) Describe how a CSRF attack works and how to test for CSRF vulnerabilities. Explain by using an example.
f) (5%) What are software security touchpoints, and how do these influence the software development process?

Problem 2 – (40 points)

Case description:
Chronic obstructive pulmonary disease (COPD), also known as chronic obstructive lung disease (COLD), chronic obstructive airway disease (COAD), in Norwegian, Kronisk obstruktiv lungesykdom (KOLS), among others, is a type of obstructive lung disease characterized by chronically poor airflow. It typically worsens over time. The main symptoms include shortness of breath, cough, and sputum production.

Patients with KOLS should daily be able to register simple data about their health situation and condition to allow follow-up by the health care system. Examples of data could be the current airflow, spit colour, exercises, etc. The patient data use a mobile device like a smart phone or pad to register the data. The information is then transmitted through a cloud solution to the hospital. The cloud solution collects data from all patients. The data is then transmitted to the local hospital. The local hospital has a separate patient database/health journal system located on a separate health network.

a) (5%) Describe the attack surface of this solution.
b) (5%) Identify and describe security threats (unwanted security incident) and possible threat agents/attackers to this solution.
c) (10%) Make a risk assessment of the solution using RMF
d) (5%) How would you mitigate the threats and risks?

e) (10%) State which security requirements and other security issues you believe should be addressed in the implementation of the solution.

f) (5%) Discuss any ethical concerns that should be taken into account in this solution?

Problem 3 – (20 points)

For each of the code snippets listen below, your task is to:

- Identify the main security vulnerabilities/issues.
- Explain why this is a security issue.
- Fix the code. You may use pseudo-code for this. Remember to explain your solution.

Code snippet 1

```
String GenerateReceiptURL(String baseUrl) {
        Random ranGen = new Random();
        ranGen.setSeed((new Date()).getTime());
        return(baseUrl + ranGen.nextInt(400000000) + ".html");
}
```

Code snippet 2

```
protected void doPost (HttpServletRequest req,
                                        HttpServletResponse res)
                        throws IOException {
        String ip = req.getRemoteAddr();
        InetAddress addr = InetAddress.getByName(ip);
        ...
        out.println("hello " + addr.getHostName());
}
```

Code snippet 3
login.php:

```
<?php
session_start();
?>
<html>
 <body>
<?php
if (isset($_SESSION[“user”])) {
    echo “<p>Welcome back, “ . $_SESSION[“user”] . “!<br>”;
    echo ‘<a href=”process.php?action=logout”>Logout</a></p>’;
}
else {
?>
  <form action=”process.php?action=login” method=”post”>
   <p>The username is: admin</p>
   <input type=”text” name=”user” size=”20”>
   <p>The password is: test</p>
   <input type=”password” name=”pass” size=”20”>
   <input type=”submit” value=”Login”>
  </form>
<?php
}
?>
 </body>
</html>
```

process.php:

```
<?php
session_start();

switch($_GET["action"]) {
    case "login":
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $user = (isset($_POST["user"]) &&
                ctype_alnum($_POST["user"]) ? $_POST["user"] : null;
            $pass = (isset($_POST["pass"])) ? $_POST["pass"] : null;
            $salt = '$2a$07$my.s3cr3t.SalTY.str1nG;

            if (isset($user, $pass) && (crypt($user . $pass, $salt) ==
                crypt("admintest", $salt))) {
                $_SESSION["user"] = $_POST["user"];
            }
        }
        break;

    case "logout":
        $_SESSION = array();
        session_destroy();
        break;
}

header("Location: login.php");
?>
```

Code snippet 4

```
class Example
{
   private $hook;

   function __construct()
   {
      // some PHP code...
   }

   function __wakeup()
   {
      if (isset($this->hook)) eval($this->hook);
   }
}

// some PHP code...

$user_data = unserialize($_COOKIE['data']);

// some PHP code...
```