SOFTWARE SECURITY

SAFE AND SECURE SOFTWARE SYSTEMS
BTU COTTBUS-SENFTENBERG

WINTER TERM
2018/2019

Helke

Publication: 28.11.18
Submission: 07.01.19
*Time to complete this exercise **about five** weeks!*

Exercise Sheet 1: Security Analyses and Information Flows

Security Analyses

A private health insurance company (PHIC) plans to use software-based administration for its business processes even more than before.

The company would like to use a content management system (CMS) for this purpose. A CMS makes it possible to provide large amounts of data from different user groups in different representations. It also supports the modification and administration of this data by authorized users.

The company plans to offer the following new features.

1. The PHIC's website is to be improved in such a way that certain statistical company data, such as the current number of policyholders or the average reimbursement of premiums to policyholders etc., can be displayed in a way that is effective for advertising purposes. This information is not only to be presented as static content, but also updated dynamically.

2. Policyholders should be able to submit invoices (doctor's bills, prescription fees, etc.) via the portal and be informed about the status of the processing of the claimed costs.

3. Hospitals should be able to bill treatment costs directly via the Internet.

4. Local service shops of health insurance companies that recruit new customers or help customers to solve problems should be able to access customer data directly from their terminals and change it if necessary.

5. Finally, PHIC, which belongs to a large insurance group, wants to pass on data about its policyholders to other companies in the group so that they can advertise new products to policyholders.

These functions are to be implemented by means of a CMS, which internally accesses the central database of the PHIC, prepares data according to their purpose and presents them to the various user groups in a suitable way.

Conduct a security analysis on at least four, but not more than five pages. Note that the security analysis does not have to be complete because of the page restriction. Nevertheless, the following artifacts of a security analysis should be included to document your analysis in an exemplary manner.

- Network topology model of the system
- Use case model to describe actors and system functionality including protection goals for each actor
- Two damage scenarios including an evaluation resulting in the required level of protection
- At least one misuse case diagram and one attack tree to refine an attacker goal
- An extended version of the attack tree, tagged by fictive costs and probabilities

In addition to the models to be created, you should also answer important questions in your analysis. Below are some sample questions that you could discuss. However, please note that not all questions are relevant for every application.

- Security Objectives
  - Which persons are involved in which roles with the system?
  - Who has what protection goals?
  - Which goals of different parties are increasing?
  - Which goals of different parties are contradictory?
- Attacker Model
  - Who can act as an attacker against the protection goals of which parties?
  - What is the interest of these attackers?
  - What power, what capabilities do these attackers have?

Information Flow Analyses

Implement multiple programs to bypass the information flow analysis of a given compiler. The programming language supported by the compiler has reduced means of expression. You are allowed to use assignments to variables, while loops, if-then-else constructs, and some other statements. Please read the documentation under the URL `http://ifc-challenge.appspot.com/`. There are 12 Boolean variables available for each program. Of these Boolean variables, 6 have a low security class and 6 have a high security class. The contents of all low variables are displayed after executing a program.

If your programs succeed in converting confidential information from the $high$ variables into low variables, you have bypassed the information flow analysis of the compiler. The compiler has different levels of information flow analysis, of which you should bypass at least five. Detailed explanations of this task are available at `http://ifc-challenge.appspot.com/`.

JIF: Information Flow Analyses for Java

JIF [1] can be used to define *Security Policies* at the source code level of Java. For example, you can use a security policy to specify which outputs are allowed on the screen and which are not. For this purpose, objects of the classes `PrintStream` and `Runtime` must be created and tagged with appropriate security labels, which take the calling *principal* into account.

Complete the given JIF code[2] in such a way that the content of a local variable x can be output to the calling *Principal Bob*. Then set the permissions so that the content of x can only be read by *Alice* and an output to *Bob* is denied.

Finally, bypass the security model of JIF using a hidden channel by adjusting the program's runtime behavior so that information about the contents of the secret variable x can transmit to *Bob* indirectly.

General Instructions

The exercises should be solved in group work (approx. 4 to 5 people). A PDF document is expected as a solution for the first subtask. For the second and third subtasks, you submit all the program codes you have created and for the second subtask, you also submit the code words determined. Please do not forget to include the title page with the names of the group members.

Solutions shall be submitted no later than 7 January 2019.

[1] `http://www.cs.cornell.edu/jif/`

[2] `https://www.informatik.tu-cottbus.de/~helke/swsec/Test.jif`