IMT3501, software security, excercise #02

Jan Samuelsen Lindemann, 120926

September 17, 2014

Abstract

This article describes issues with race condition in a electronic id solution .

1 What is a race condition

As described in "Operativsystemer"-course last year, with Erik Hjelms, a race condition occurs when 2 threads/processes are accessing the same resource, modifying it and storing it again. If this process is not implemented correctly, the results can vary greatly, depending on thread scheduling algoritms running.

2 Race conditions in a electronic ID solution

I have thought up 3 different cases of a possible race condition when dealing with a electronic ID solution, for instance logging in to a website or online service.

2.1 Race condition 1

2.1.1 condition

If the service allows the user to log in on multiple systems(laptop, mobile phone, tablet, etc) and keeps track of the users state by something as simple as a "logged in" binary-switch of some sort, a possible race condition could be:

1. User logs in on his iPad, switch gets set to "logged in"
2. User uses service on his iPad
3. User logs in on his mobile phone, switch gets set to "logged in"(even though its alread set to it.)
4. User logs off from iPad, switch gets set to "logged off"
5. User continues to try using the service on his mobile phone, to his surprise, he is logged off.

2.1.2 solution

Either keep track of the users state on each individual device, or only allow one device to be logged in at any given time.

2.2 Race condition 2

2.2.1 condition

Lets say the service fixed the above mentioned case and allowed users to be logged in on multiple devices. The user shares the account on the service with his friend/wife/whatever.

1. 1. user logs in on his pc.
2. 2. user tries to log on to the service on a different pc.
3. 1. user wants to change the login-credentials(username and password)
4. while 2. users credentials are being checked, the changes that the 1. user requested are being implemented.
5. Either 2. user is rejected(because he/she was a bit too slow) or the 2. user might end up with wrong information while logged in?

2.2.2 solution

Don't allow changing of login-information unless there is only 1 user logged in.

2.3 Race condition 3

2.3.1 condition

Because the user keeps fucking up the system with the above cases, the administrators have decided that from now on, he is only allowed to read stuff from the service. However, the following happens:

1. User logs in
2. User decides he wants to change some value of some sensitive system-parameter
3. User types in the commands to get the prompt to change the parameter
4. Administrator implements the changes, that dont allow the user to change anything anymore.
5. User enters the new value, presses enter.
6. Since the user typed in the command to change the parameters before the changes to his rights where implemented, the user's changes are accepted since the system thinks he has the authority to do so.

2.3.2 solution

Check for authority when initiatiing change, and afterwards, when the system finally is accepting the changes.