Obligatory exercise 02
Race conditions

Jan Kerkenhoff
141628
IMT3501, Software Security

September 17, 2014

1 Introduction

For this Assignment we need to think of race conditions in an electronic Id solution, I choose an electronic passenger identification system for trains or metros to find race conditions. This system is responsible to control access to the trains via gates where a card need to swiped.

1.1 Definition of a race condition

Race conditions occur when two process access a shared resource in parallel without proper synchronization. This leads to corrupted data sets or unwanted behavior

2 Race condition 1

If you image a system which is connected to the main servers via some kind of connection. At every gate the system need to check if the person traveling has a valid ticket for this type of train. Every time someone swipes his card at the terminal the system checks the ticket type on the card and signals if entering is allowed. The system offers a card type that offers one time entry and then travel for up to one hour. To make the system faster especially during rush hours the information, about the time the user entered, is saved to the system after the user is already in the system. This type of card can be exploited as a possible race condition depending on the time it takes the system to communicate with the server. If you swipe the card at the terminal, get through and immediately pass it to the next person, which also passes it through the terminal. Since the terminal checks at the server for a entry date with this card and then opens the gate. The following problem arises:

```
Terminal process 1 checks at the server for entry date
Terminal process 1 gets returned null and the gate opens
Terminal process 1 starts to add the entry date to the server
Terminal process 2 checks at the server for entry date
Terminal process 2 gets returned null and the gate opens
Terminal process 2 starts to add the entry date to the server
Terminal process 1 finishes adding the entry date to the server
Terminal process 2 finishes adding the entry date to the server
```

This race condition allows as many people access to the system as you can fit in the time it takes the terminal to write to the server. Same way should allow everyone to leave the system if the software is setup in the same way.

The Fix To counter this kind of race condition there is a simple solution. All the Terminals should use an internal cache of the accepted cards while the updates to the server have not finished. This would deny the use of the card after the first time and therefore disallow multiple people on one card.

3 Race condition 2

The system has a promotional website where you register for one free ride on the trains. This system uses the users data to ensure everyone only gets one card. The User enters his email and address and then gets send an card for the train via mail. The system checks if already a card was sent to this address and denies any further request if it finds one. Every time you hit the submit button a request is send to the server, your data stays in place and no session management is used. That's where we can uses a race condition to our advantage. If we supply the server with a wast amount of requests at exact the same time every request will be processed simultaneously and all would validate true, since there is no prior entry for this address. The system would send out multiple cards to the attackers address.

The Fix Countering this attack is very simple, just use a session management of some kind and disable multiple clicks of the submit button. Now only one request per session is possible.