Assignment 2

BY Synne Gran Østern – 120922

In this assignment I will imagine an electronic ID solution, consider and explain 2 possible race conditions and their effects and finally describe some countermeasures for this race conditions.

Race Conditions

Race conditions happens when two processes simultaneously want to use the same resources at the same time. The consequences will therefore be deleterious for one or both processes. This is something the attackers might want to exploit without the legitimate users knowledge. Furthermore, Dowd uses examples in the UNIX file system code. If a process enters a blocking system call at an inopportune moment, you might get a race condition. A inopportune moment happens usually during a "sensitive multiple-step operation" in a file or directory manipulation. When the other process gets the resources during these sensitive steps, it can subvert the nonatomic sequence and force the adversary process' privileges. [1]

Electronic ID solution – Bank Customer Electronic Identification

The imaginary solution I will discuss is an electronic identification for bank to identify their online customer. I would like to emphasize that this solution is not the Norwegian "BankID" solution, but very similar. The Bank Customer Electronic Identification (I will follow the ICT tradition and simplify by referring to this as "BCEI") is a mechanism to identify unique customer for a bank. The customer logs on their respective "homepage" for their bank relation (shows eg their accounts, loans, payments etc) by entering their username and chosen password. When this is checked by the banks customer database, and a match is found, the customer will get approved the logon-session and automatically redirected to their bank homepage. Furthermore, if the customer wish to transfer money from an account to another account in the same bank, an identification session is begun. The customer enters the account number, amount and name, before the identification session begins. The ID-session get checked in the bank database and if a match is found, the transaction is enabled.

Race Conditions

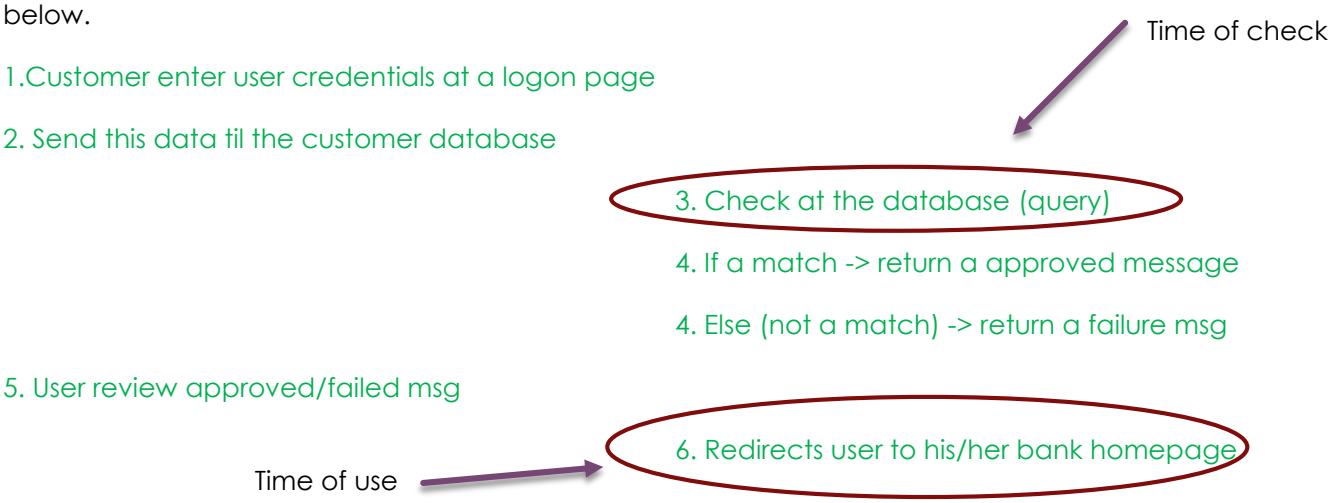
The solution described above is a very simple and from a security perspective not very secure, but it is the intention to simply to make it clearer what race conditions it might face. In my opinion there are 2 critical sessions that can meet race conditions. I will explain them separately.

---

[1]Mark Dowd, John McDonald og Justin Schuh. *The Art of software security.* Addison-Wesley. 5ed. Nov.2013.

Logon Session

As mention in the explanation of the solution above, the logon session consist of the customers interaction by entering their username and password at a logon page, and the check at the bank customer database. I have written a pseudo code of how this communication happens below.



This logon session may be vulnerable for the TOCTTOU issue. The TOCTTOU stands for "Time Of check To Time Of Use" and is exploiting the discrepancy between the query (security check) and the use of the resource (each customer bank homepage). [2] To exploit this "loophole" the attackers must time its attack to perfection, but it is possible to be successful in this exploit. The consequences of this might be that the attackers (or an ignorant customer) access the original customer bank home page. I believe this might be the "worst case" scenario, both for the customer, but also for the bank. Other possible consequences might be that the customer do not get access to his/her bank homepage at all.

Transaction session

Another situation that can be vulnerable to a race condition is the transaction session. This is when the customer, who have legitimately logged on correctly and are at his/her bank homepage, and want to transfer money to another customers account. The pseudo code below shows how this transaction happens.

Customer	Bank
1. Customer enters the bank account number, amount and name to where *he wish to transfer to.	
	2. Bank checks the bank account nr and name(query) 3. If a match (the bank account exist) opens a transaction object. 4. Return a "ok" msg to customer

[2] Mark Dowd, John McDonald og Justin Schuh. *The Art of software security.* Addison-Wesley. 5ed. Nov.2013.

5. ID-session of the customer (enters username and password)

6. When ID-session is successful, enable transaction

Step 2 is the "time of check" and step 6 is the "time of use". Between these two steps, there are possibilities to have a TOCTTOU issue. An attacker in this situation may misuse the transaction object to change account number or amount. The customer will not be able to detect this change because right after the ID-session of the customer, the transaction is enabled and the money is transferred from the victims account to modified account number or event pay a different amount that originally entered.

Countermeasures

As mentioned above, this bank and its BCEI, does not incorporate security in a very successfully way. But what can you do to mitigate race conditions and specially the TOCTTOU issue?

In the SANS Institute InfoSec Reading Rooms *A tour of TOCTTOUs* they refer the "**Immutable bindings"** as a solution to mitigate TOCTTOU. A immutable binding use a static reference to make sure that it is always the same object that works on the resource. If there are already a reference that are static in the existing solution, this would be the best to use. [3] However, this is not the case with this solution. Therefore, we have to create one. This will work the same way as a hardware based semaphore. The steps between the check and use must happen in one atomic operation. We will have to create a critical sector for the sessions mention above. [4]

Other solutions the same article mentions is: "**Increase the number of checks"** and "**Check nearer the use".** But neither of these will not eliminate the possibly to exploit the TOCTTOU issue, only reduce the probability.[5]

[3] J. Craig Lowery, The SANS Institute InfoSec Reading Room. *A tour of TOCTTOU*. August 2002. URL: http://www.sans.org/reading-room/whitepapers/securecode/tour-tocttous-1049
[4] Se above
[5] Se above