Thread Safety Problems

Saturday, October 21, 2017 2:45 PM

- This exploits a concurrency error

- Web applications can handle many HTTP requests simultaneously
- Developers often use variables that are not thread safe
  - Thread safety means that the fields of an object or class always maintain a valid state when used concurrently by multiple threads
- It is often possible to exploit a concurrency bug by loading the same page as another user at the exact same time
- Because all threads share the same method area, and the method area is where all class variables are stored, multiple threads can attempt to use the same class variables concurrently

Solution

- Open a 2nd web browser and navigate to the same page
- In one browser enter 'jeff' in the user name field
- In the other enter 'dave'
- Click submit on the them both very quickly
- You get the same information on both pages
  - Effectively, you hijacked the account information from the other user

***The root cause of this exploit is that the Java code uses a static variable for the user name.
When submitting twice, the same thread and hence the same static variable containing the username of the first request will be used.

This is obvious when examining the Java code:

private static String currentUser;

Shopping Cart Concurrency Flaw

Solution

- Open a 2nd web browser to the same page
- In the first window select a low cost item and click Purchase
- In the 2nd window select a high cost item and click Update Cart
- The global (static) variable will be overwritten but your price variable will remain the same
- Click Confirm in the first window and you will get the high cost item from window 2nd for the price of the low price item in window 1