**Race Condition: Format Flaw Development Mitigation SOP**

Race condition vulnerabilities through format flaws occur because the methods *parse()* and *format()* in *java.text.Format* contain a design flaw that can cause one user to see another user’s data. This vulnerability comes from those methods containing a race condition.

**Defense Against Race Condition: Format Flaw**

Synchronization should be used to protect again race conditions whenever *parse()* or *format()* in *java.text.Format* is used.

**Example**

public class Common {

private static SimpleDateFormat dateFormat;

…

public String format( Date date ) {

return dateFormat.format( date );

}

…

final OtherClass dateFormatAccess = new OtherClass();

…

public void function\_running\_in\_thread1() {

System.out.println( “Time in thread 1 should be 12/31/69 4:00 PM, found ” + dateFormatAccess.format( new Date(0) ) );

}

public void function\_running\_in\_thread2() {

System.out.println( “Time in thread 2 should be around 12/29/09 6:26 AM, found: ” + dateFormatAccess.format( new Date(System.currentTimeMillis() ) ) );

}

}

Explanation:

1. The problem with the code above occurs when two users run this code simultaneously because the following output could be produced:  
     
   Time in thread 1 should be 12/31/69 4:00 PM, **found: 12/31/69 4:00 PM**  
   Time in thread 2 should be around 12/29/09 6:26 AM, **found: 12/31/69 4:00 PM**  
     
   From the bolded text above, it can be seen that the date from the first output is used in the second output (bolded above) due to a race condition in the implementation of *format()*
2. The code below demonstrates how the use of synchronization constructs can prevent race conditions
3. Another additional fix would be to the use *org.apache.commons.lang.time.FastDateFormat* class because it is the thread-safe version of *java.text.SimpleDateFormat*

public class Common {

private static SimpleDateFormat dateFormat;

…

public synchronized String format1( Date date ) {

return dateFormat.format( date );

}

public String format2( Date date ) {

synchronized( dateFormat ) {

return dateFormat.format( date );

}

}

}

**References**

[HP Enterprise Security – Race Condition: Format Flaw](http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/race_condition_format_flaw.html)