



Findings for SpotBugs, the plugin for Find Security Bugs with Cisco's eCommerce use case

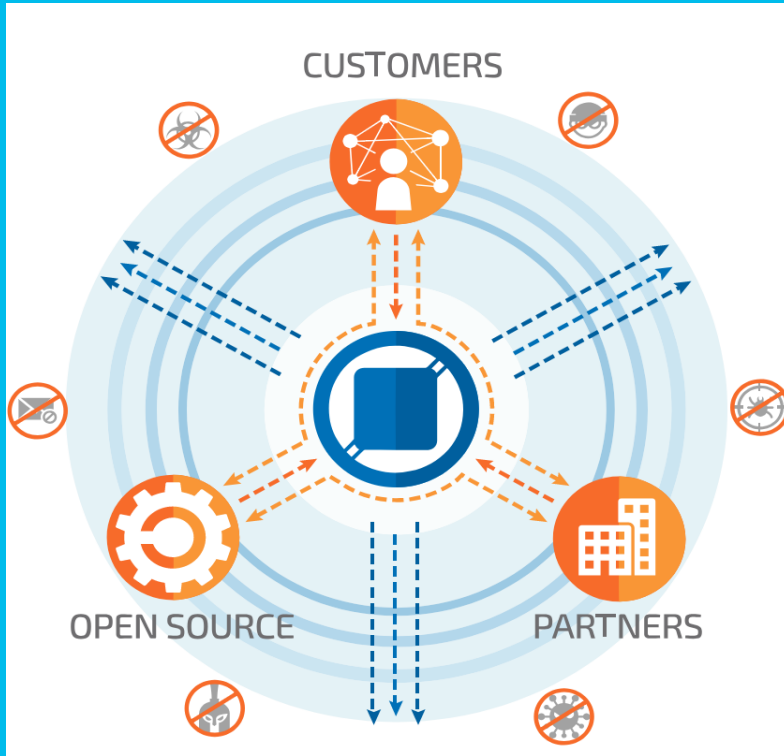
Jay Chow

Software Engineer, Cisco eCommerce, IT, Operations

Cisco, San Jose

30th July 2020

Agenda



- 1 What is Find Security Bugs?
- 2 What is SpotBugs?
- 3 Example 1: Getting hands on
- 4 Example 2: Getting hands on
- 5 Thoughts on using SpotBugs
- 6 References

What is Find Security Bugs?

- The SpotBugs plugin for security audits of Java web applications
- Features include:
 - 135 bug patterns, detecting 135 different vulnerability types: <https://find-sec-bugs.github.io/bugs.htm>
 - Continuous integration with systems such as Jenkins
 - Integration with IDE such as Eclipse and command line integration with Maven
 - OWASP TOP 10 and CWE coverage
 - Support for frameworks including Struts and Spring-MVC

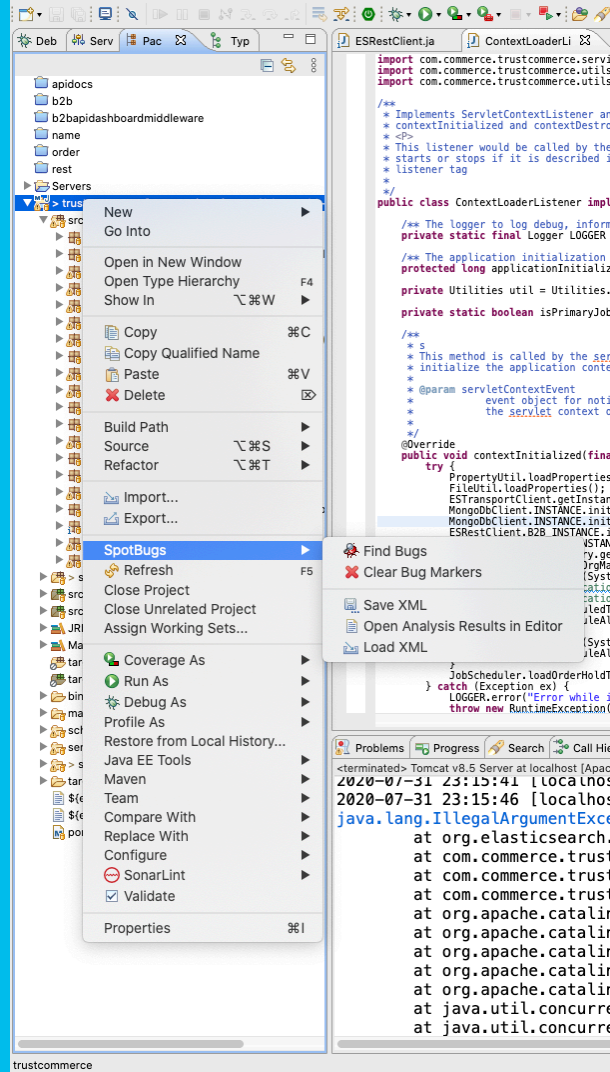
What is SpotBugs?

- Program which uses static analysis to look for bugs in Java code
- Successor of FindBugs
- Requires JRE (or JDK) 1.8.0 or later to run but it can analyze programs compiled for any version of Java, from 1.0 to 1.9
- SpotBugs checks for more than 400 bug patterns, with descriptions found: <https://spotbugs.readthedocs.io/en/latest/bugDescriptions.html>



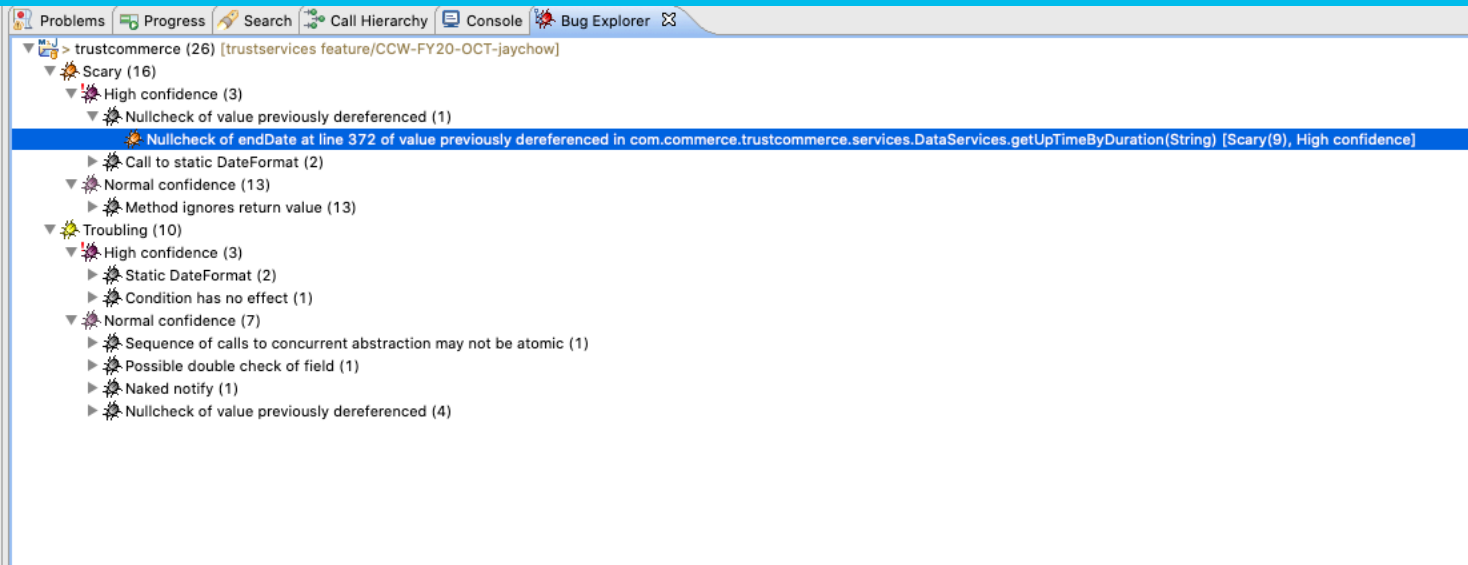
Getting hands-on with SpotBugs

- 1) Used Eclipse marketplace to install Eclipse plugin
- 2) Help > Eclipse Marketplace > SpotBugs
- 3) After installing, right click on project > SpotBugs > Find Bugs



Example 1: Getting hands-on with SpotBugs

4) Go to Bug Explorer and double click on the line:



5) Click on the bug icon on the left side:

372 `LocalDateTime midNightTime = LocalDateTime.of(endDate.toLocalDate(), midnight);`

Example 1: Getting hands-on with SpotBugs

6) Click on the bug icon which will bring you Bug Info:

ProblemsProgressSearchCall HierarchyConsoleBug ExplorerBug Info

DataServices.java: 372

Navigation

Nullcheck of endDate at line 372 of value previously dereferenced in com.commerce.trustcommerce.services.DataServices.getUpTimeByDuration(String)
Value loaded from endDate
Return value of com.commerce.trustcommerce.io.ServiceRequest.getEndDate() of type java.time.LocalDateTime
Redundant null check at DataServices.java:[line 375]
Redundant null check at DataServices.java:[line 375]

Bug: Nullcheck of endDate at line 372 of value previously dereferenced in com.commerce.trustcommerce.services.DataServices.getUpTimeByDuration(String)

A value is checked here to see whether it is null, but this value can't be null because it was previously dereferenced and if it were null a null pointer exception would have occurred at the earlier dereference. Essentially, this code and the previous dereference disagree as to whether this value is allowed to be null. Either the check is redundant or the previous dereference is erroneous.

Rank: Scary (9), **confidence:** High
Pattern: RCN_REDUNDANT_NULLCHECK_WOULD_HAVE_BEEN_A_NPE
Type: RCN, **Category:** CORRECTNESS (Correctness)

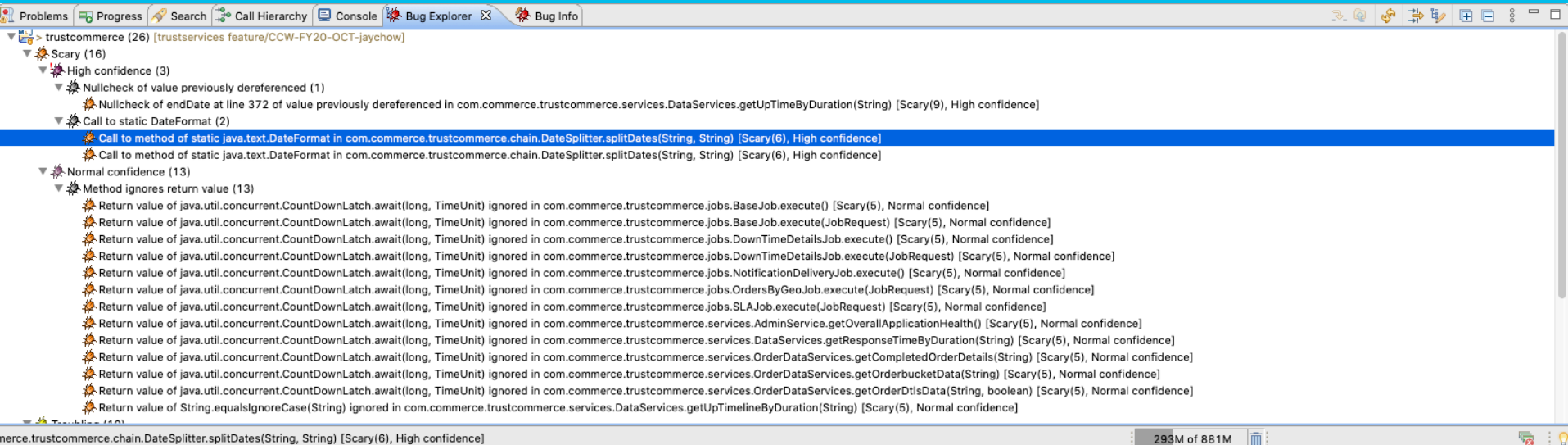
XML output:

```
<BugInstance type="RCN_REDUNDANT_NULLCHECK_WOULD_HAVE_BEEN_A_NPE" priority="1" rank="9" abbrev="RCN" category="CORRECTNESS" first="1">
  <Class classname="com.commerce.trustcommerce.services.DataServices">
    <SourceLine classname="com.commerce.trustcommerce.services.DataServices" sourcefile="DataServices.java" sourcepath="com/commerce/trustcommerce/services/DataServices.java"/>
  </Class>
  <Method classname="com.commerce.trustcommerce.services.DataServices" name="getUpTimeByDuration" signature="(Ljava/lang/String;)Ljava/lang/String;" isStatic="false">
    <SourceLine classname="com.commerce.trustcommerce.services.DataServices" start="356" end="500" startBytecode="0" endBytecode="3808" sourcefile="DataServices.java" sourcepath="com/commerce/trustcommerce/services/DataServices.java"/>
  </Method>
  <LocalVariable name="endDate" register="10" pc="125" role="LOCAL_VARIABLE_VALUE_OF"/>
  <Method classname="com.commerce.trustcommerce.io.ServiceRequest" name="getEndDate" signature="()Ljava/time/LocalDateTime;" isStatic="false" role="METHOD_RETURN_VALUE_OF">
    <SourceLine classname="com.commerce.trustcommerce.io.ServiceRequest" start="8" end="8" startBytecode="0" endBytecode="46" sourcefile="ServiceRequest.java" sourcepath="com/commerce/trustcommerce/io/ServiceRequest.java"/>
  </Method>
  <SourceLine classname="com.commerce.trustcommerce.services.DataServices" start="372" end="372" startBytecode="105" endBytecode="105" sourcefile="DataServices.java" sourcepath="com/commerce/trustcommerce/services/DataServices.java"/>
  <SourceLine classname="com.commerce.trustcommerce.services.DataServices" start="375" end="375" startBytecode="127" endBytecode="127" sourcefile="DataServices.java" sourcepath="com/commerce/trustcommerce/services/DataServices.java" rol
  <SourceLine classname="com.commerce.trustcommerce.services.DataServices" start="375" end="375" startBytecode="127" endBytecode="127" sourcefile="DataServices.java" sourcepath="com/commerce/trustcommerce/services/DataServices.java" rol
</BugInstance>
```

Note: bug pattern is STCAL_INVOKE_ON_STATIC_DATE_FORMAT_INSTANCE

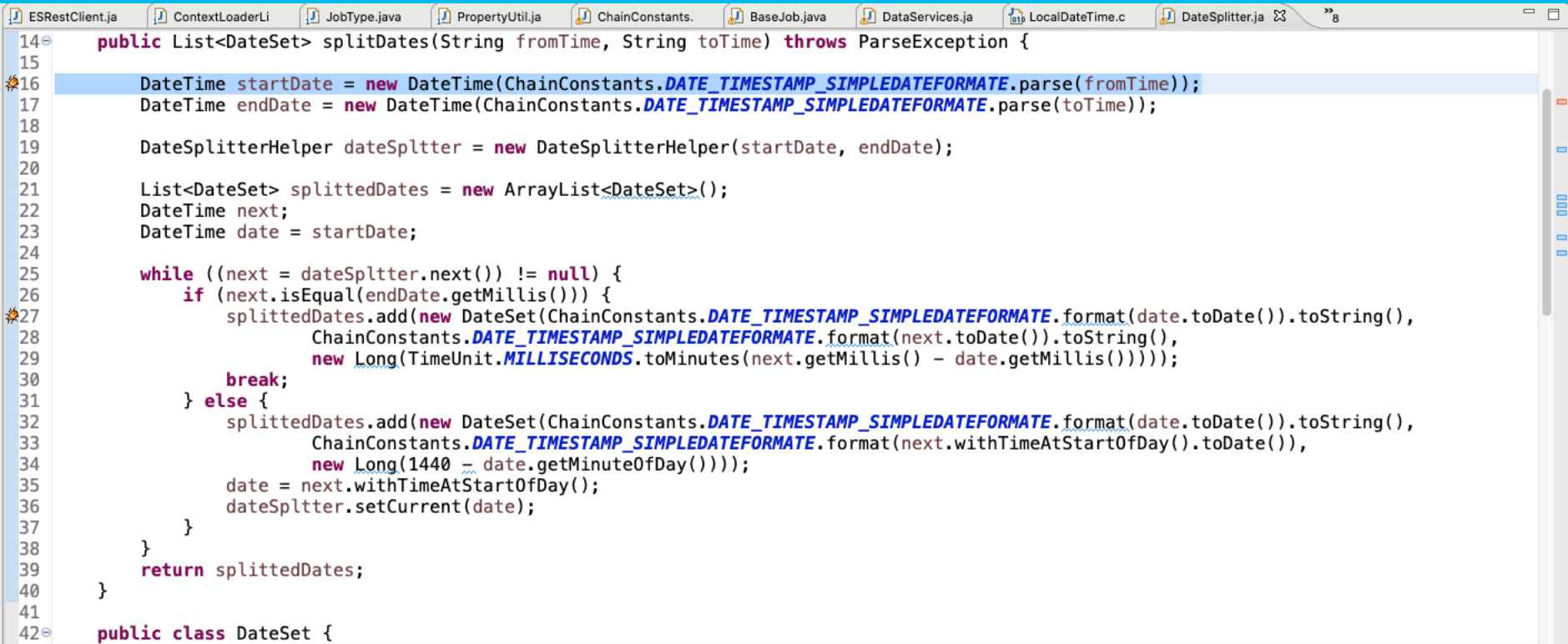
Example 2: Getting hands-on with SpotBugs

1) Under Bug Explorer:



Example 2: Getting hands-on with SpotBugs

2) After clicking on the bug identified by SpotBugs on previous page:

A screenshot of an IDE window showing a Java file named DateSplitter.java. The code defines a method splitDates that takes fromTime and toTime strings and returns a List of DateSet objects. A SpotBugs bug is highlighted on line 27, indicating a 'Null Pointer Dereference' on the variable 'next'. The IDE interface includes a tab bar at the top with several open files, a left margin with line numbers, and a right sidebar with icons for project structure and search.

```
14 public List<DateSet> splitDates(String fromTime, String toTime) throws ParseException {
15
16     DateTime startDate = new DateTime(ChainConstants.DATE_TIMESTAMP_SIMPLEDATEFORMAT.parse(fromTime));
17     DateTime endDate = new DateTime(ChainConstants.DATE_TIMESTAMP_SIMPLEDATEFORMAT.parse(toTime));
18
19     DateSplitterHelper dateSplttter = new DateSplitterHelper(startDate, endDate);
20
21     List<DateSet> splittedDates = new ArrayList<DateSet>();
22     DateTime next;
23     DateTime date = startDate;
24
25     while ((next = dateSplttter.next()) != null) {
26         if (next.isEqual(endDate.getMillis())) {
27             splittedDates.add(new DateSet(ChainConstants.DATE_TIMESTAMP_SIMPLEDATEFORMAT.format(date.toDate()).toString(),
28                 ChainConstants.DATE_TIMESTAMP_SIMPLEDATEFORMAT.format(next.toDate()).toString(),
29                 new Long(TimeUnit.MILLISECONDS.toMinutes(next.getMillis() - date.getMillis()))));
30             break;
31         } else {
32             splittedDates.add(new DateSet(ChainConstants.DATE_TIMESTAMP_SIMPLEDATEFORMAT.format(date.toDate()).toString(),
33                 ChainConstants.DATE_TIMESTAMP_SIMPLEDATEFORMAT.format(next.withTimeAtStartOfDay().toDate()),
34                 new Long(1440 - date.getMinuteOfDay())));
35             date = next.withTimeAtStartOfDay();
36             dateSplttter.setCurrent(date);
37         }
38     }
39     return splittedDates;
40 }
41
42 public class DateSet {
```

Example 2: Getting hands-on with SpotBugs

3) After clicking on bug icon on the left of the line number on the previous page:

ProblemsProgressSearchCall HierarchyConsoleBug ExplorerBug Info

DateSplitter.java: 16

Navigation

Call to method of static java.text.DateFormat in com.commerce.trustcommerce.chain.DateSplitter.splitDates(String, String)
Field com.commerce.trustcommerce.chain.ChainConstants.DATE_TIMESTAMP_SIMPLEDATEFORMAT
Called method java.text.SimpleDateFormat.parse(String)

Bug: Call to method of static java.text.DateFormat in com.commerce.trustcommerce.chain.DateSplitter.splitDates(String, String)

As the JavaDoc states, DateFormats are inherently unsafe for multithreaded use. The detector has found a call to an instance of DateFormat that has been obtained via a static field. This looks suspicious.

For more information on this see [JDK Bug #6231579](#) and [JDK Bug #6178997](#).

Rank: Scary (6), **confidence:** High
Pattern: STCAL_INVOKE_ON_STATIC_DATE_FORMAT_INSTANCE
Type: STCAL, **Category:** MT_CORRECTNESS (Multithreaded correctness)

XML output:

```
<BugInstance type="STCAL_INVOKE_ON_STATIC_DATE_FORMAT_INSTANCE" priority="1" rank="6" abbrev="STCAL" category="MT_CORRECTNESS" first="1">
<Class classname="com.commerce.trustcommerce.chain.DateSplitter">
  <SourceLine classname="com.commerce.trustcommerce.chain.DateSplitter" sourcefile="DateSplitter.java" sourcepath="com/commerce/trustcommerce/chain/DateSplitter.java"/>
</Class>
<Method classname="com.commerce.trustcommerce.chain.DateSplitter" name="splitDates" signature="(Ljava/lang/String;Ljava/lang/String;)Ljava/util/List;" isStatic="false">
  <SourceLine classname="com.commerce.trustcommerce.chain.DateSplitter" start="16" end="39" startBytecode="0" endBytecode="567" sourcefile="DateSplitter.java" sourcepath="com/commerce/trustcommerce/chain/DateSplitter.java"/>
</Method>
<Method classname="java.text.SimpleDateFormat" name="parse" signature="(Ljava/lang/String;)Ljava/util/Date;" isStatic="false" role="METHOD_CALLED">
  <SourceLine classname="java.text.SimpleDateFormat" sourcefile="SimpleDateFormat.java" sourcepath="java/text/SimpleDateFormat.java"/>
</Method>
<Field classname="com.commerce.trustcommerce.chain.ChainConstants" name="DATE_TIMESTAMP_SIMPLEDATEFORMAT" signature="Ljava/text/SimpleDateFormat;" isStatic="true">
  <SourceLine classname="com.commerce.trustcommerce.chain.ChainConstants" sourcefile="ChainConstants.java" sourcepath="com/commerce/trustcommerce/chain/ChainConstants.java"/>
</Field>
<SourceLine classname="com.commerce.trustcommerce.chain.DateSplitter" start="16" end="16" startBytecode="8" endBytecode="8" sourcefile="DateSplitter.java" sourcepath="com/commerce/trustcommerce/chain/DateSplitter.java"/>
<SourceLine classname="com.commerce.trustcommerce.chain.DateSplitter" start="16" end="16" startBytecode="8" endBytecode="8" sourcefile="DateSplitter.java" sourcepath="com/commerce/trustcommerce/chain/DateSplitter.java"/>
</BugInstance>
```

Note: bug pattern is STCAL_INVOKE_ON_STATIC_DATE_FORMAT_INSTANCE

Example 2: Getting hands-on with SpotBugs

4) After clicking on JDK Bug #6178997:

ORACLE[®] Java Bug Database

Search

Oracle Technology Network > Java > Java SE > Community > Bug Database

JDK-6178997 : (cal) Doc: it's not explicitly documented java.util.Calendar serialization are thread-unsafe

Type: Bug	Priority: P4	Submitted: 2004-10-14
Component: core-libs	Status: Open	Updated: 2017-02-10
Sub-Component: java.util:i18n	Resolution: Unresolved	
Affected Version: 1.4.2	OS: linux, windows_2000	
	CPU: x86	

Description

FULL PRODUCT VERSION :
java version "1.4.2_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_01-b06)
Java HotSpot(TM) Client VM (build 1.4.2_01-b06, mixed mode)

ADDITIONAL OS VERSION INFORMATION :

Thoughts on using Spotbugs...

- Intuitively integrated with Eclipse IDE via marketplace
- Can be used standalone and through integrations including Ant, Maven and Gradle
- Reliable with good online documentation at <https://spotbugs.readthedocs.io/en/stable/>
- Learned that confidence measures the likelihood that SpotBugs has flagged a real bug
- Recommended to use as secondary tool during java development for java-based applications/microservices across Cisco as false positives may occur
- Comprehensive list of descriptions and coding solutions given for bugs patterns identified by the plugin: <https://find-sec-bugs.github.io/bugs.htm>
- Overall a pretty robust tool

References

<https://find-sec-bugs.github.io/>

<https://github.com/find-sec-bugs/find-sec-bugs>

<https://spotbugs.github.io/>

<https://github.com/spotbugs/spotbugs>