ATTACK DEFENSE

by PentesterAcademy

Name	FindSecBugs: Securing Java Applications
URL	attackdefense.com/challengedetails?cid=2050
Type	DevSecOps Basics: Automated Code Review

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

This setup consists of a Kali machine with FindSecBugs and a Gitlab instance.

Challenge Description

FindSecBugs is a Spotbugs plugin that is used to perform security audits of Java web applications and Android applications. It is also available as a CLI utility.

A Kali GUI machine (kali-gui) is provided to the user with findsecbugs installed on it. The source code of a sample web application is provided in the home directory of the root user.

Objective: Use FindSecBugs to find issues in the code.

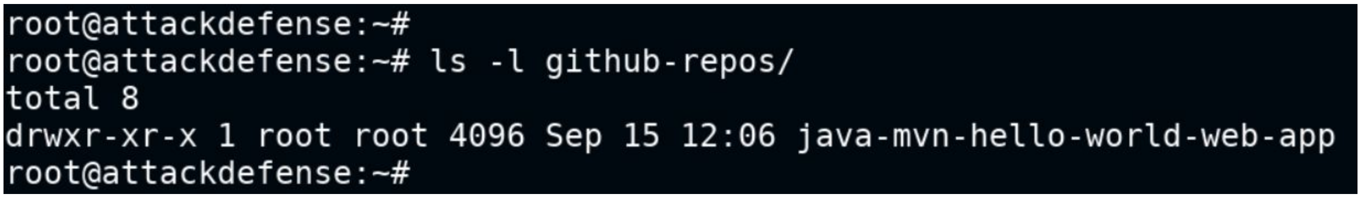
Instructions:

- The source code of web application is provided at /root/github-repos

Solution

Step 1: Open the terminal and check the provided web application.

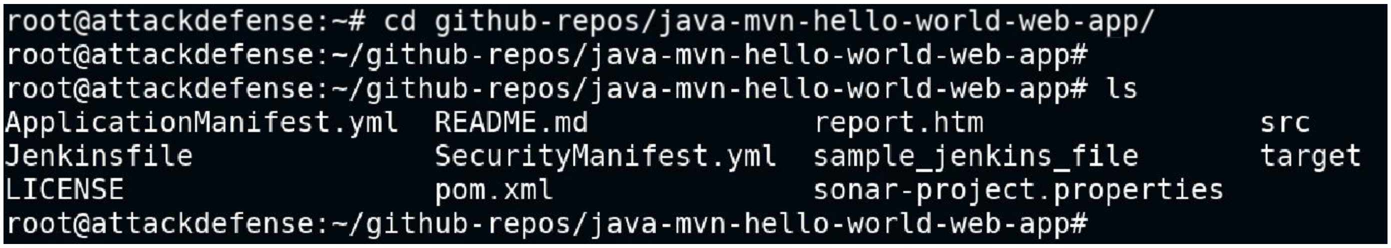
Command: ls -l github-repos



```
root@attackdefense:~#
root@attackdefense:~# ls -l github-repos/
total 8
drwxr-xr-x 1 root root 4096 Sep 15 12:06 java-mvn-hello-world-web-app
root@attackdefense:~#
```

Step 2: Change to the cloned directory and check its contents.

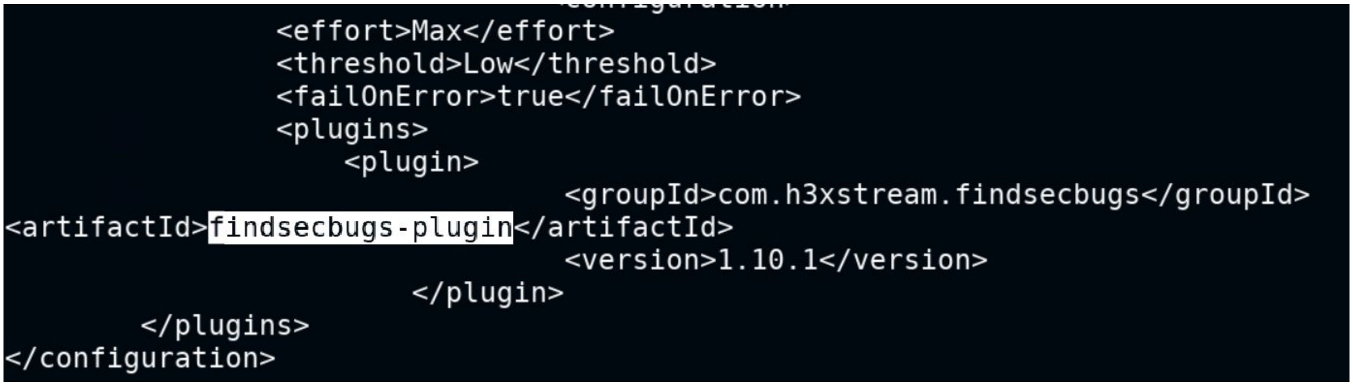
Commands:
cd github-repos/java-mvn-hello-world-web-app
ls



```
root@attackdefense:~# cd github-repos/java-mvn-hello-world-web-app/
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app# ls
ApplicationManifest.yml  README.md              report.htm                 src
Jenkinsfile              SecurityManifest.yml   sample_jenkins_file        target
LICENSE                  pom.xml                sonar-project.properties
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
```

Step 3: Check the contents of pom.xml of the application. The POM (Project Object Model) file contains the configuration information for the maven to build the project.

Command: cat pom.xml

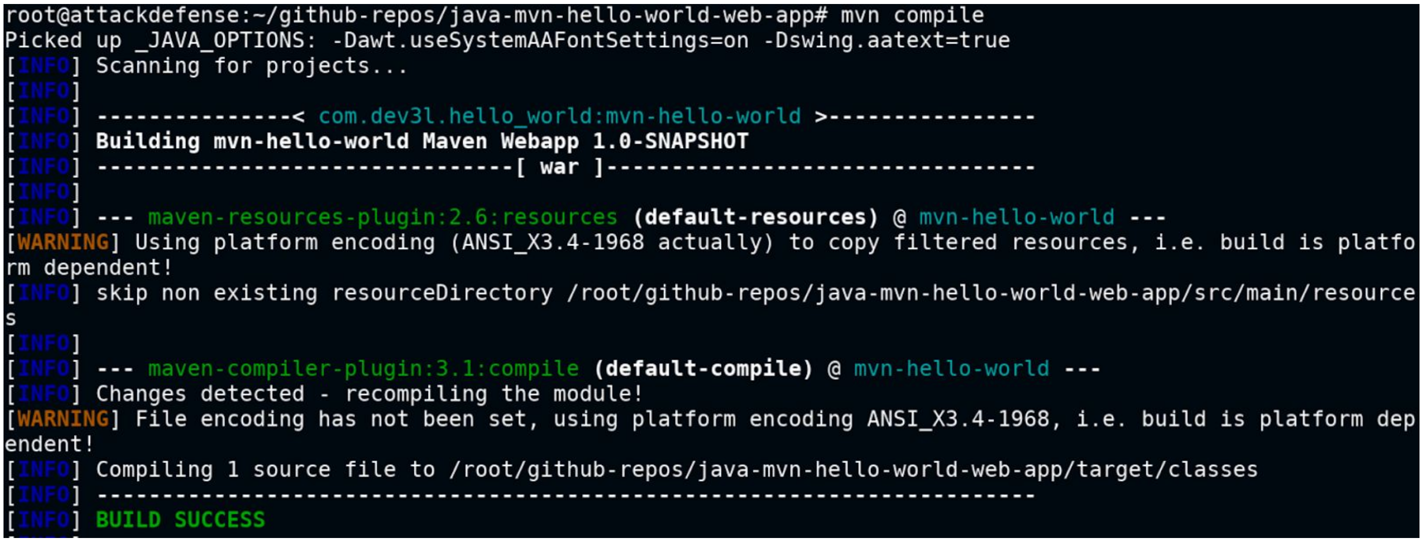


```
                <effort>Max</effort>
                <threshold>Low</threshold>
                <failOnError>true</failOnError>
                <plugins>
                    <plugin>
                                 <groupId>com.h3xstream.findsecbugs</groupId>
<artifactId>findsecbugs-plugin</artifactId>
                                 <version>1.10.1</version>
                        </plugin>
        </plugins>
</configuration>
```

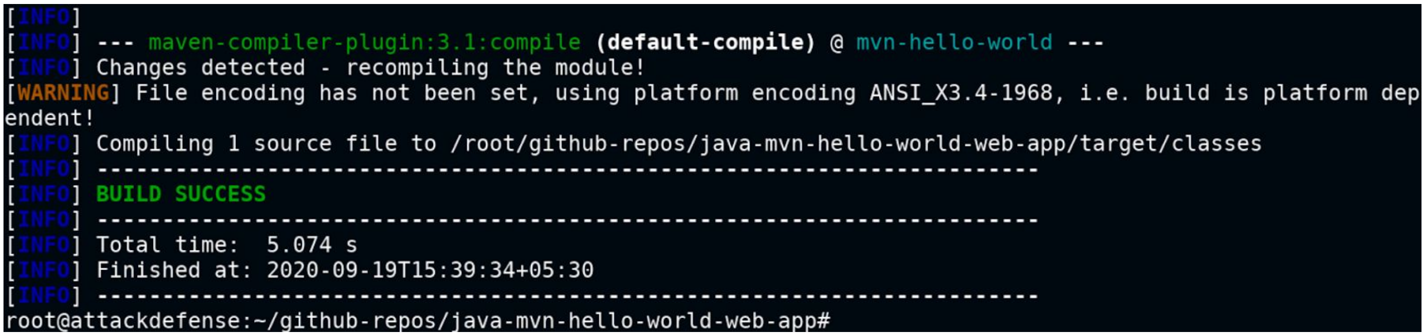
The pom.xml contains the configuration to install findsecbugs plugin in the application.

Step 4: Compile the Java code.

Command: mvn compile



```
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app# mvn compile
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[INFO] Scanning for projects...
[INFO]
[INFO] ---------------< com.dev3l.hello_world:mvn-hello-world >----------------
[INFO] Building mvn-hello-world Maven Webapp 1.0-SNAPSHOT
[INFO] --------------------------------[ war ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ mvn-hello-world ---
[WARNING] Using platform encoding (ANSI_X3.4-1968 actually) to copy filtered resources, i.e. build is platfo
rm dependent!
[INFO] skip non existing resourceDirectory /root/github-repos/java-mvn-hello-world-web-app/src/main/resource
s
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mvn-hello-world ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding ANSI_X3.4-1968, i.e. build is platform dep
endent!
[INFO] Compiling 1 source file to /root/github-repos/java-mvn-hello-world-web-app/target/classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
```



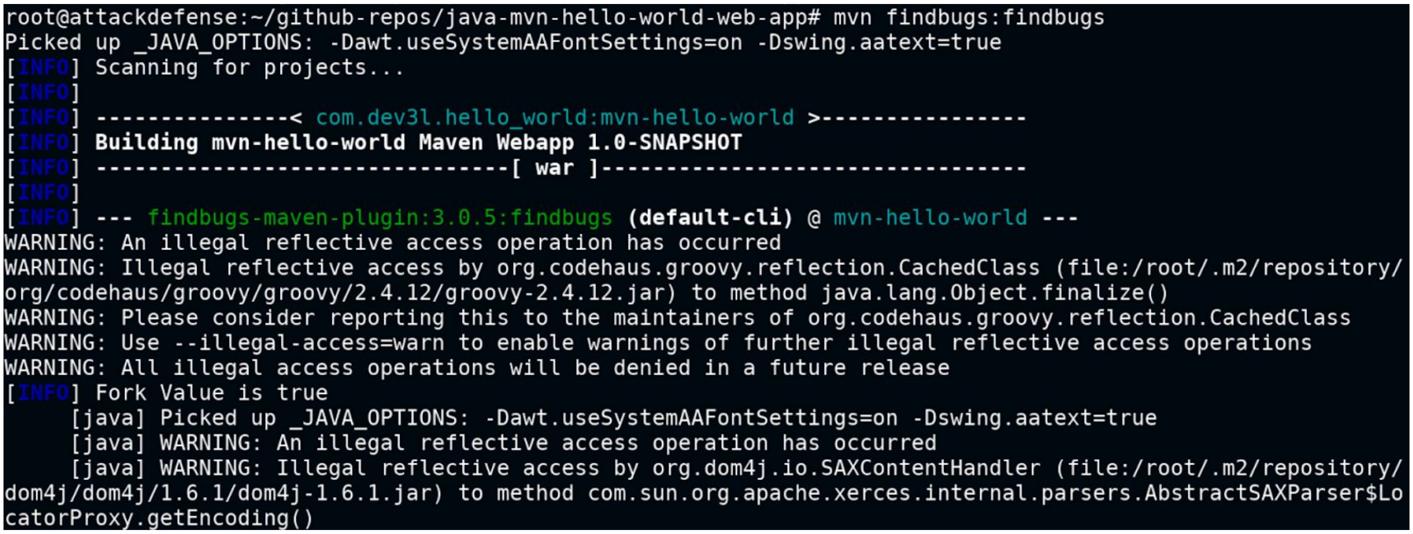
```
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mvn-hello-world ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding ANSI_X3.4-1968, i.e. build is platform dep
endent!
[INFO] Compiling 1 source file to /root/github-repos/java-mvn-hello-world-web-app/target/classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
```

```
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.074 s
[INFO] Finished at: 2020-09-19T15:39:34+05:30
[INFO] ------------------------------------------------------------------------
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
```

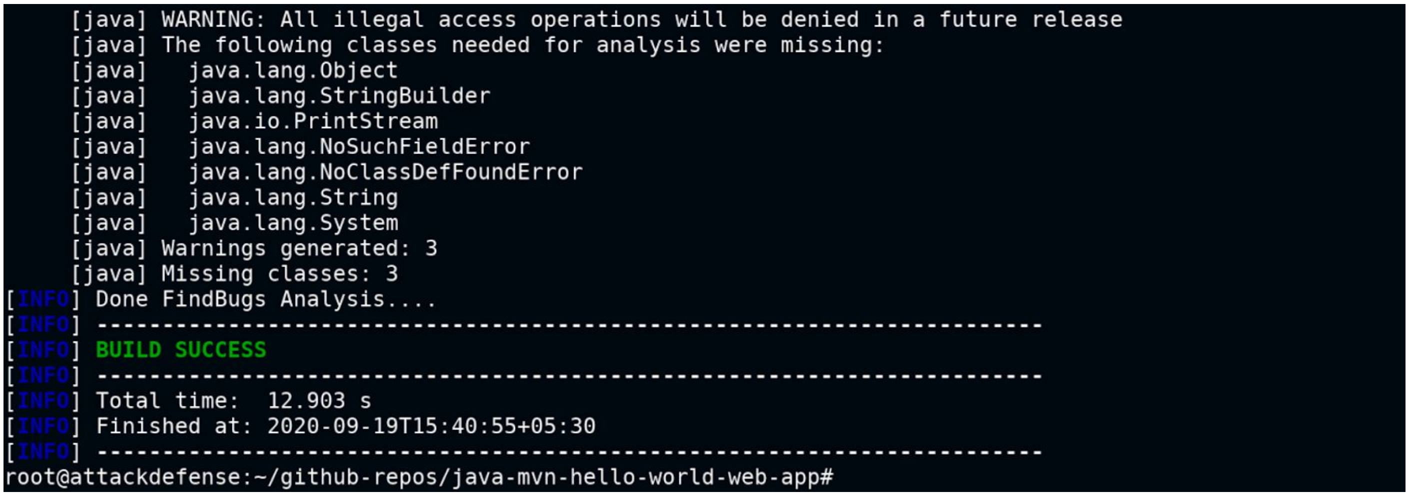
Maven will compile the source code and build the application. After building the application, findsecbugs can be used to scan for vulnerabilities.

Step 5: Now, we need to install the packages for findsecbugs and scan the application with it.

Command: mvn findbugs:findbugs



```
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app# mvn findbugs:findbugs
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[INFO] Scanning for projects...
[INFO]
[INFO] ---------------< com.dev3l.hello_world:mvn-hello-world >----------------
[INFO] Building mvn-hello-world Maven Webapp 1.0-SNAPSHOT
[INFO] --------------------------------[ war ]---------------------------------
[INFO]
[INFO] --- findbugs-maven-plugin:3.0.5:findbugs (default-cli) @ mvn-hello-world ---
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.codehaus.groovy.reflection.CachedClass (file:/root/.m2/repository/
org/codehaus/groovy/groovy/2.4.12/groovy-2.4.12.jar) to method java.lang.Object.finalize()
WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.reflection.CachedClass
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Fork Value is true
     [java] Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
     [java] WARNING: An illegal reflective access operation has occurred
     [java] WARNING: Illegal reflective access by org.dom4j.io.SAXContentHandler (file:/root/.m2/repository/
dom4j/dom4j/1.6.1/dom4j-1.6.1.jar) to method com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser$Lo
catorProxy.getEncoding()
```

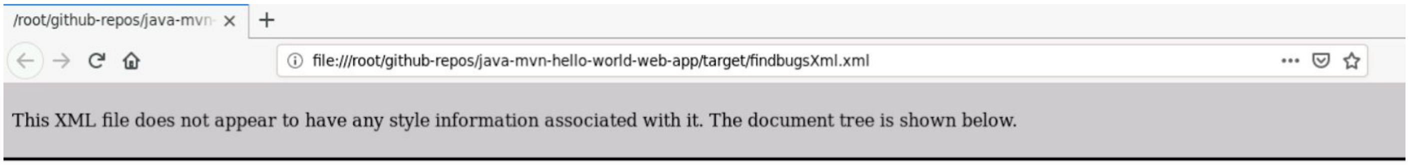


```
     [java] WARNING: All illegal access operations will be denied in a future release
     [java] The following classes needed for analysis were missing:
     [java]   java.lang.Object
     [java]   java.lang.StringBuilder
     [java]   java.io.PrintStream
     [java]   java.lang.NoSuchFieldError
     [java]   java.lang.NoClassDefFoundError
     [java]   java.lang.String
     [java]   java.lang.System
     [java] Warnings generated: 3
     [java] Missing classes: 3
[INFO] Done FindBugs Analysis....
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  12.903 s
[INFO] Finished at: 2020-09-19T15:40:55+05:30
[INFO] ------------------------------------------------------------------------
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
```

The step got executed successfully. This will also generate a report in the target directory (/root/github-repos/java-mvn-hello-world-web-app/target/findbugsXml.xml).

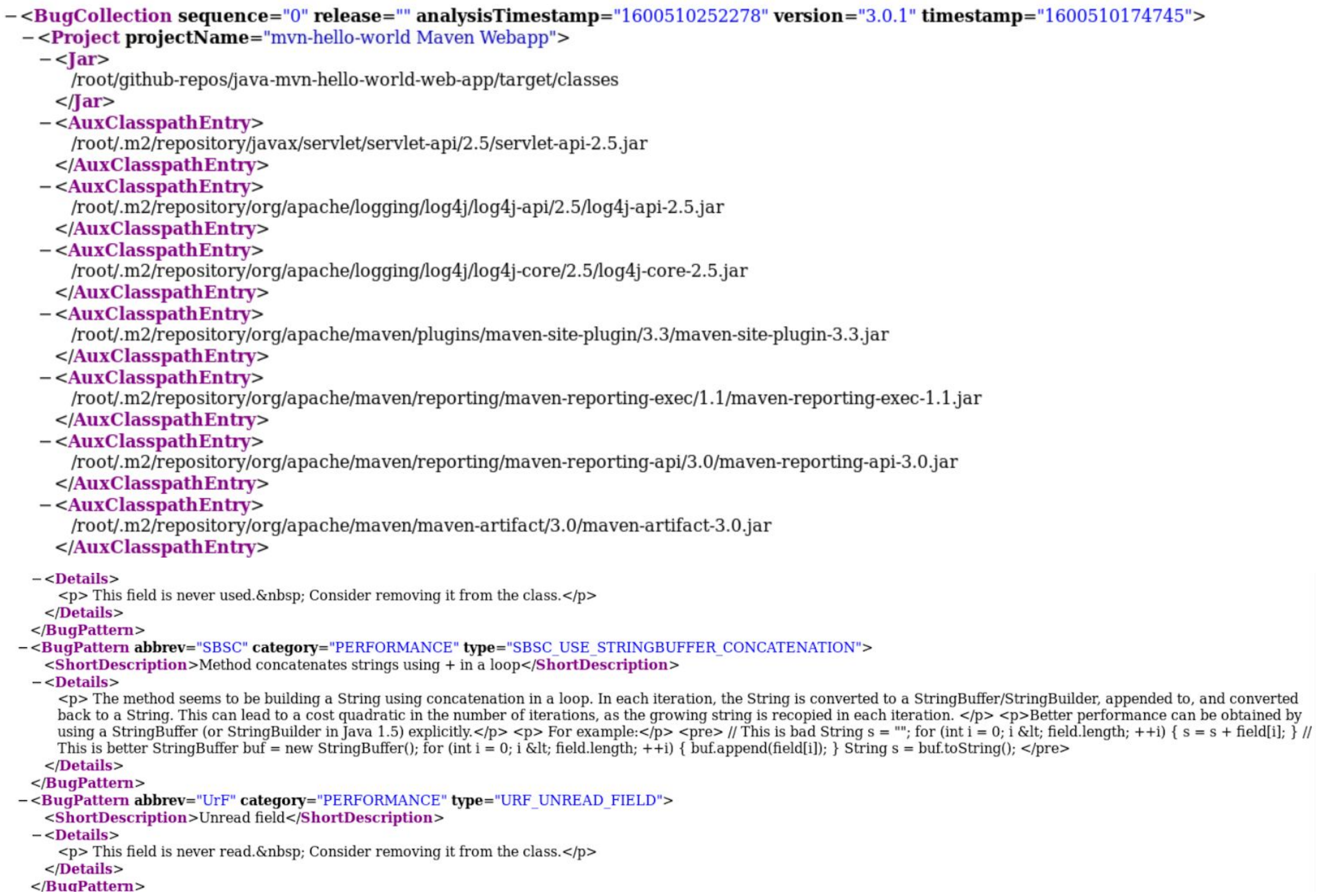
Step 6: Check the generated report.

Command: firefox target/findbugsXml.xml



file:///root/github-repos/java-mvn-hello-world-web-app/target/findbugsXml.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.



```
- <BugCollection sequence="0" release="" analysisTimestamp="1600510252278" version="3.0.1" timestamp="1600510174745">
  - <Project projectName="mvn-hello-world Maven Webapp">
    - <Jar>
        /root/github-repos/java-mvn-hello-world-web-app/target/classes
      </Jar>
    - <AuxClasspathEntry>
        /root/.m2/repository/javax/servlet/servlet-api/2.5/servlet-api-2.5.jar
      </AuxClasspathEntry>
    - <AuxClasspathEntry>
        /root/.m2/repository/org/apache/logging/log4j/log4j-api/2.5/log4j-api-2.5.jar
      </AuxClasspathEntry>
    - <AuxClasspathEntry>
        /root/.m2/repository/org/apache/logging/log4j/log4j-core/2.5/log4j-core-2.5.jar
      </AuxClasspathEntry>
    - <AuxClasspathEntry>
        /root/.m2/repository/org/apache/maven/plugins/maven-site-plugin/3.3/maven-site-plugin-3.3.jar
      </AuxClasspathEntry>
    - <AuxClasspathEntry>
        /root/.m2/repository/org/apache/maven/reporting/maven-reporting-exec/1.1/maven-reporting-exec-1.1.jar
      </AuxClasspathEntry>
    - <AuxClasspathEntry>
        /root/.m2/repository/org/apache/maven/reporting/maven-reporting-api/3.0/maven-reporting-api-3.0.jar
      </AuxClasspathEntry>
    - <AuxClasspathEntry>
        /root/.m2/repository/org/apache/maven/maven-artifact/3.0/maven-artifact-3.0.jar
      </AuxClasspathEntry>

    - <Details>
        <p> This field is never used.&nbsp; Consider removing it from the class.</p>
      </Details>
    </BugPattern>
  - <BugPattern abbrev="SBSC" category="PERFORMANCE" type="SBSC_USE_STRINGBUFFER_CONCATENATION">
      <ShortDescription>Method concatenates strings using + in a loop</ShortDescription>
    - <Details>
        <p> The method seems to be building a String using concatenation in a loop. In each iteration, the String is converted to a StringBuffer/StringBuilder, appended to, and converted
        back to a String. This can lead to a cost quadratic in the number of iterations, as the growing string is recopied in each iteration. </p> <p>Better performance can be obtained by
        using a StringBuffer (or StringBuilder in Java 1.5) explicitly.</p> <p> For example:</p> <pre> // This is bad String s = ""; for (int i = 0; i &lt; field.length; ++i) { s = s + field[i]; } //
        This is better StringBuffer buf = new StringBuffer(); for (int i = 0; i &lt; field.length; ++i) { buf.append(field[i]); } String s = buf.toString(); </pre>
      </Details>
    </BugPattern>
  - <BugPattern abbrev="UrF" category="PERFORMANCE" type="URF_UNREAD_FIELD">
      <ShortDescription>Unread field</ShortDescription>
    - <Details>
        <p> This field is never read.&nbsp; Consider removing it from the class.</p>
      </Details>
    </BugPattern>
```

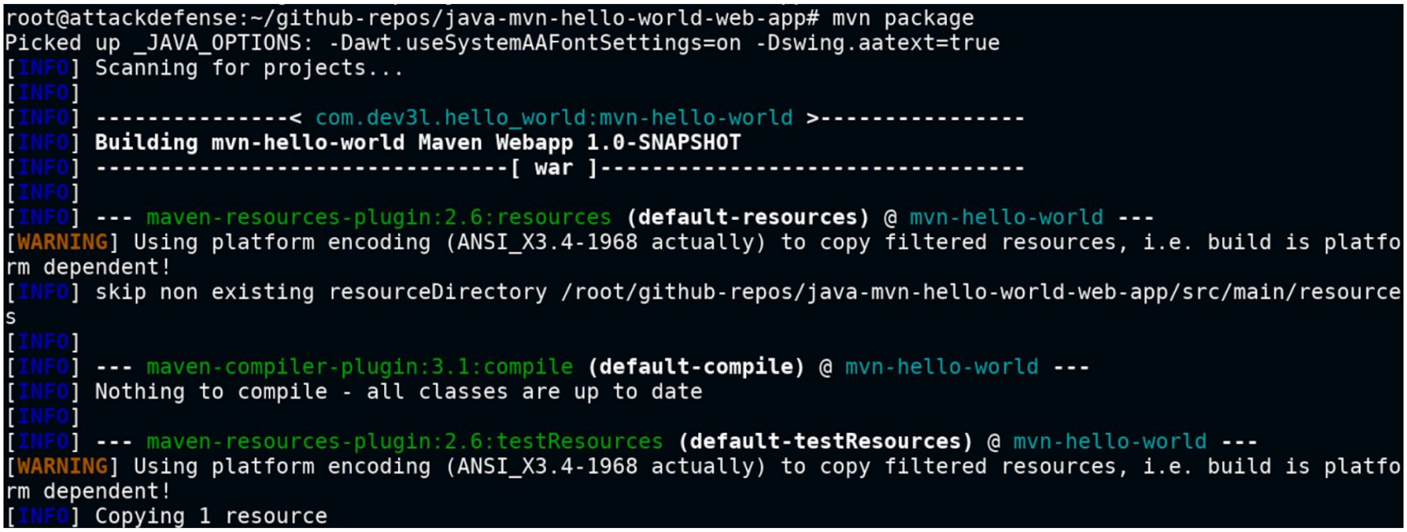
Issues Detected:

- Plus (+) operator is used for concatenation
- Unread field in the code

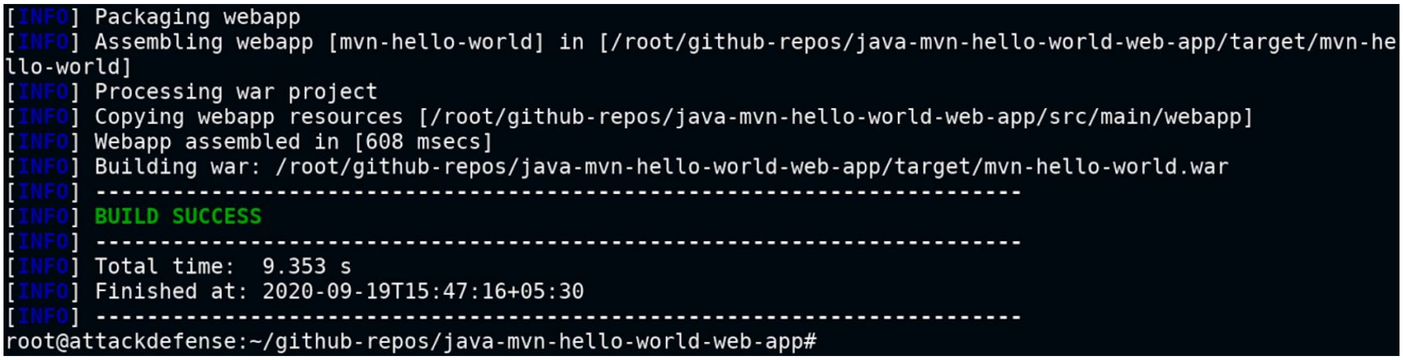
Note: There were other issues too in the application report, we have trimmed the XML file in this screenshot.

Step 7: Package the application using maven. The application will now be packaged into a single standalone jar file.

Command: mvn package



```
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app# mvn package
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[INFO] Scanning for projects...
[INFO]
[INFO] ---------------< com.dev3l.hello_world:mvn-hello-world >----------------
[INFO] Building mvn-hello-world Maven Webapp 1.0-SNAPSHOT
[INFO] --------------------------------[ war ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ mvn-hello-world ---
[WARNING] Using platform encoding (ANSI_X3.4-1968 actually) to copy filtered resources, i.e. build is platfo
rm dependent!
[INFO] skip non existing resourceDirectory /root/github-repos/java-mvn-hello-world-web-app/src/main/resource
s
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mvn-hello-world ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ mvn-hello-world ---
[WARNING] Using platform encoding (ANSI_X3.4-1968 actually) to copy filtered resources, i.e. build is platfo
rm dependent!
[INFO] Copying 1 resource
```

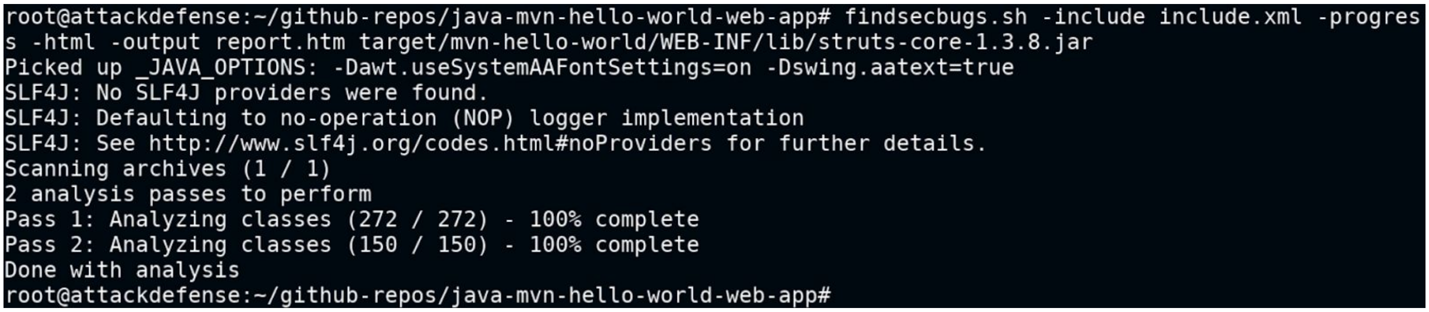


```
[INFO] Packaging webapp
[INFO] Assembling webapp [mvn-hello-world] in [/root/github-repos/java-mvn-hello-world-web-app/target/mvn-he
llo-world]
[INFO] Processing war project
[INFO] Copying webapp resources [/root/github-repos/java-mvn-hello-world-web-app/src/main/webapp]
[INFO] Webapp assembled in [608 msecs]
[INFO] Building war: /root/github-repos/java-mvn-hello-world-web-app/target/mvn-hello-world.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  9.353 s
[INFO] Finished at: 2020-09-19T15:47:16+05:30
[INFO] ------------------------------------------------------------------------
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
```

The packaging was successful. The jar file is stored in the target directory.

Step 8: Utilise the FindSecBugs CLI tool to scan the application.

Command: findsecbugs.sh -include include.xml -progress -html -output report.htm target/mvn-hello-world/WEB-INF/lib/struts-core-1.3.8.jar

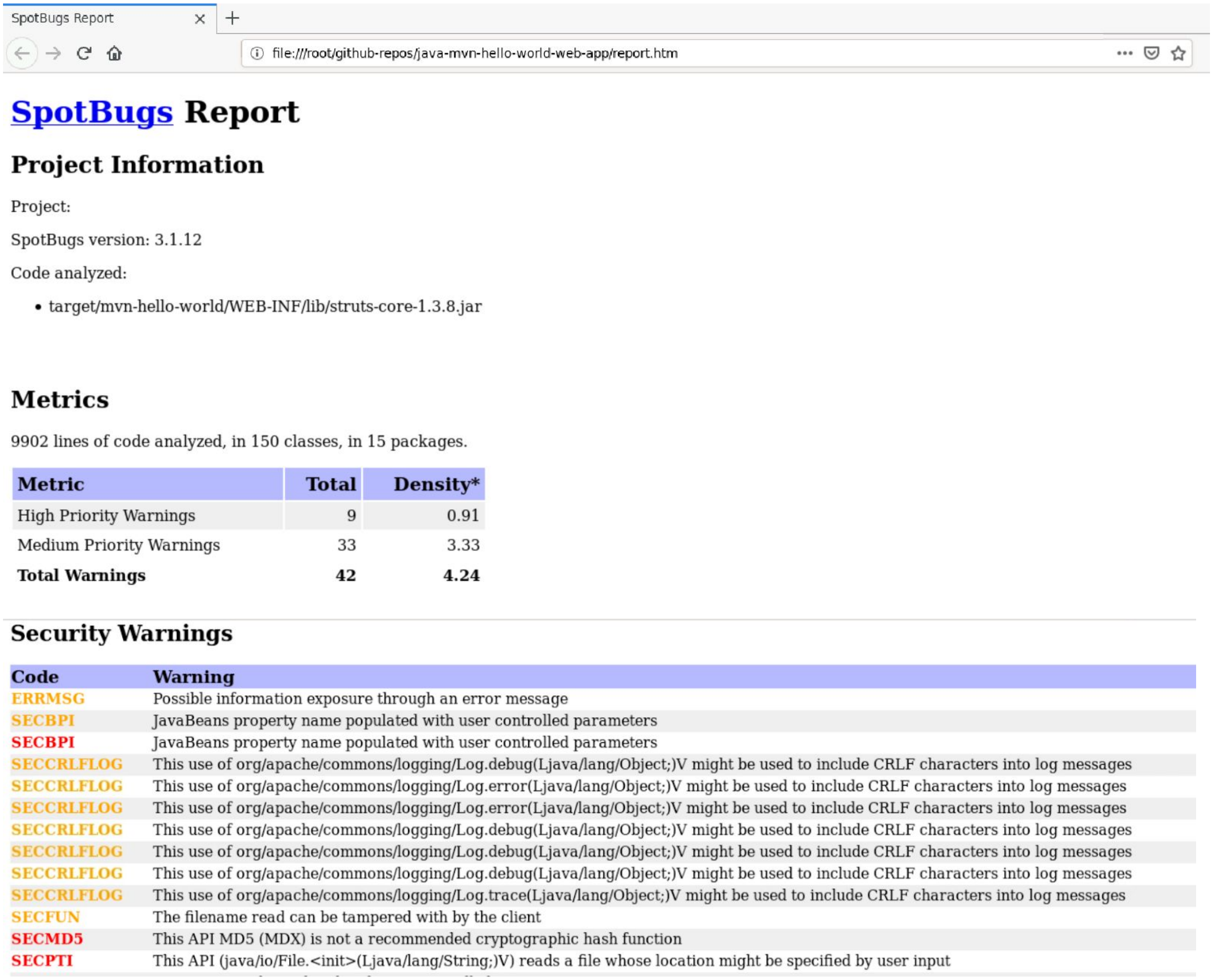


```
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app# findsecbugs.sh -include include.xml -progres
s -html -output report.htm target/mvn-hello-world/WEB-INF/lib/struts-core-1.3.8.jar
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#noProviders for further details.
Scanning archives (1 / 1)
2 analysis passes to perform
Pass 1: Analyzing classes (272 / 272) - 100% complete
Pass 2: Analyzing classes (150 / 150) - 100% complete
Done with analysis
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
```

The scan was completed and a report.htm file was generated after the scan.

Step 9: Open the report.htm file in firefox (/root/github-repos/java-mvn-hello-world-web-app/report.htm).

Command: firefox report.htm



SpotBugs Report
file:///root/github-repos/java-mvn-hello-world-web-app/report.htm

SpotBugs Report

Project Information

Project:

SpotBugs version: 3.1.12

Code analyzed:

- target/mvn-hello-world/WEB-INF/lib/struts-core-1.3.8.jar

Metrics

9902 lines of code analyzed, in 150 classes, in 15 packages.

Metric	Total	Density*
High Priority Warnings	9	0.91
Medium Priority Warnings	33	3.33
Total Warnings	**42**	**4.24**

Security Warnings

Code	Warning
ERRMSG	Possible information exposure through an error message
SECBPI	JavaBeans property name populated with user controlled parameters
SECBPI	JavaBeans property name populated with user controlled parameters
SECCRLFLOG	This use of org/apache/commons/logging/Log.debug(Ljava/lang/Object;)V might be used to include CRLF characters into log messages
SECCRLFLOG	This use of org/apache/commons/logging/Log.error(Ljava/lang/Object;)V might be used to include CRLF characters into log messages
SECCRLFLOG	This use of org/apache/commons/logging/Log.error(Ljava/lang/Object;)V might be used to include CRLF characters into log messages
SECCRLFLOG	This use of org/apache/commons/logging/Log.debug(Ljava/lang/Object;)V might be used to include CRLF characters into log messages
SECCRLFLOG	This use of org/apache/commons/logging/Log.debug(Ljava/lang/Object;)V might be used to include CRLF characters into log messages
SECCRLFLOG	This use of org/apache/commons/logging/Log.debug(Ljava/lang/Object;)V might be used to include CRLF characters into log messages
SECCRLFLOG	This use of org/apache/commons/logging/Log.trace(Ljava/lang/Object;)V might be used to include CRLF characters into log messages
SECFUN	The filename read can be tampered with by the client
SECMD5	This API MD5 (MDX) is not a recommended cryptographic hash function
SECPTI	This API (java/io/File.<init>(Ljava/lang/String;)V) reads a file whose location might be specified by user input

Issues Detected

- Information Exposure
- Weak Algorithm Used (md5)

Learnings

Perform security audits using the findsecbugs tool.