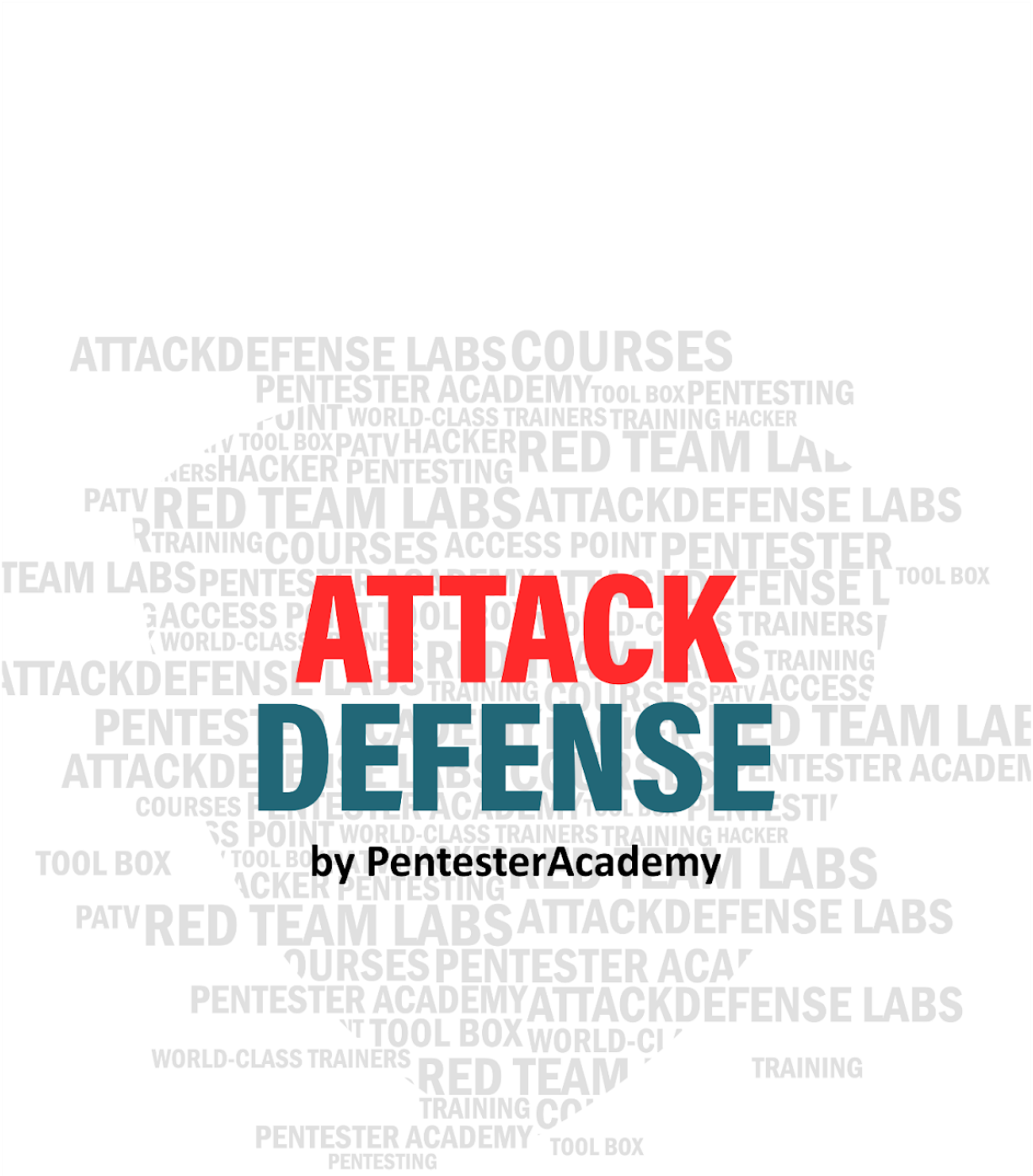
ATTACK
DEFENSE
by PentesterAcademy

Name	PMD: Finding Common Vulnerabilities
URL	https://www.attackdefense.com/challengedetails?cid=2049
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.

Challenge Description

PMD is a source code analyzer that is used to find common vulnerabilities in the source code (Java, Javascript, etc) for example, empty cache blocks or unused variables.

A Kali CLI machine (kali-cli) is provided to the user with PMD installed on it. The source code for two sample web applications is provided in the home directory of the root user.

Objective: Analyze the given source code with PMD to find security issues!

Instructions:

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

Solution

Step 1: Check the provided web applications.

Command: ls -l github-repos

```
root@attackdefense:~# ls -l github-repos/
total 8
drwxrwxr-x 7 root root 4096 Sep 14 05:57 gradle-simple
drwxrwxr-x 5 root root 4096 Sep 14 05:57 java-mvn-hello-world-web-app
root@attackdefense:~#
```

We will take one example at a time and run the tool on that.

Example 1: java maven hello world web app

Step 1: Change to the java-maven-hello-world-web-app 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  LICENSE  README.md           SecurityManifest.yml      src
Jenkinsfile              pom.xml  sample_jenkins_file  sonar-project.properties  target
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
```

Step 2: 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

```
        <plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-pmd-plugin</artifactId>
<version>3.0.1</version>
<executions>
<execution>
                <goals>
                <goal>check</goal>
                </goals>
</execution>
</executions>
        </plugin>
```

The pom.xml dictates that PMD will be used as a plugin with maven system.

Step 3: Compile the source code with maven.

Command: mvn clean compile

```
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app# mvn clean compile
[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-clean-plugin:2.5:clean (default-clean) @ mvn-hello-world ---
[INFO] Deleting /root/github-repos/java-mvn-hello-world-web-app/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ mvn-hello-world ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform depend
ent!
[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 UTF-8, i.e. build is platform dependent!
[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:  5.270 s
[INFO] Finished at: 2020-09-19T17:38:54Z
[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 pmd can be used to scan for vulnerabilities.

Step 4: Start scanning the source code using the PMD plugin. PMD plugin can be triggered with the goal ‘check’ which was defined in the POM.xml under ‘maven-pmd-plugin’ section.

Command: mvn pmd:check

```
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app# mvn pmd:check
[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-pmd-plugin:3.0.1:check (default-cli) > :pmd @ mvn-hello-world >>>
[INFO]
[INFO] --- maven-pmd-plugin:3.0.1:pmd (pmd) @ mvn-hello-world ---
[WARNING] Unable to locate Source XRef to link to - DISABLED
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO]
[INFO] <<< maven-pmd-plugin:3.0.1:check (default-cli) < :pmd @ mvn-hello-world <<<
[INFO]
[INFO]
[INFO] --- maven-pmd-plugin:3.0.1:check (default-cli) @ mvn-hello-world ---
```

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.389 s
[INFO] Finished at: 2020-09-19T17:40:08Z
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-pmd-plugin:3.0.1:check (default-cli) on projec
t mvn-hello-world: You have 1 PMD violation. For more details see:/root/github-repos/java-mvn-hello-world-we
b-app/target/pmd.xml -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
```

The check failed as the PMD plugin has found a violation.

Step 5: Check the contents of the report generated to get the details for violation.

Command: cat target/pmd.xml

```
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app# cat target/pmd.xml
<?xml version="1.0" encoding="UTF-8"?>
<pmd version="5.0.2" timestamp="2020-09-19T17:40:07.751">
<file name="/root/github-repos/java-mvn-hello-world-web-app/src/main/java/com/webapp/examples/App.java">
<violation beginline="4" endline="4" begincolumn="37" endcolumn="40" rule="UnusedPrivateField" ruleset="Unus
ed Code" package="com.webapp.examples" class="App" variable="flag" externalInfoUrl="http://pmd.sourceforge.n
et/pmd-5.0.2/rules/java/unusedcode.html#UnusedPrivateField" priority="3">
Avoid unused private fields such as 'flag'.
</violation>
</file>
</pmd>
root@attackdefense:~/github-repos/java-mvn-hello-world-web-app#
```

Issues Detected

- Unused Private field (when a private field is declared and/or assigned a value, but not used)
  Read more:
  https://pmd.github.io/latest/pmd_rules_java_bestpractices.html#unusedprivatefield

Example 2: Gradle Simple

Step 1: Change to the gradle-simple directory and check its contents.

Commands:
cd ~/github-repos/gradle-simple
ls

```
root@attackdefense:~/github-repos# cd gradle-simple/
root@attackdefense:~/github-repos/gradle-simple#
root@attackdefense:~/github-repos/gradle-simple# ls
build  build.gradle  gradle  gradlew  gradlew.bat  LICENSE  README.md  src
root@attackdefense:~/github-repos/gradle-simple#
```

Step 2: Check the contents of build.gradle. The build.gradle file contains the configuration information for the project

Command: cat build.gradle

```
root@attackdefense:~/github-repos/gradle-simple# cat build.gradle
plugins {
    id 'pmd'
}


apply plugin: 'java'
apply plugin: 'maven'
```

The build.gradle dictates that PMD will be used as a plugin with maven system.

Step 3: Run the Gradle check command using gradlew (Gradle wrapper) to check the source code using PMD plugin.

Command: ./gradlew check

```
root@attackdefense:~/github-repos/gradle-simple# ./gradlew check
Starting a Gradle Daemon, 1 incompatible Daemon could not be reused, use --status for details

> Task :pmdMain
This analysis could be faster, please consider using Incremental Analysis: https://pmd.github.io/pmd-6.8.0/p
md_userdocs_incremental_analysis.html

> Task :pmdMain FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':pmdMain'.
> 2 PMD rule violations were found. See the report at: file:///root/github-repos/gradle-simple/build/reports
/pmd/main.html

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log outpu
t. Run with --scan to get full insights.
```

```
* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log outpu
t. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 17s
2 actionable tasks: 2 executed
root@attackdefense:~/github-repos/gradle-simple#
```

The check failed as the PMD plugin has found two violations.

Step 4: Check the content of the generated report and find information about the violations.

Command: cat build/reports/pmd/main.html

```
root@attackdefense:~/github-repos/gradle-simple# cat build/reports/pmd/main.html
<html><head><title>PMD</title></head><body>
<center><h3>PMD report</h3></center><center><h3>Problems found</h3></center><table align="center" cellspacin
g="0" cellpadding="3"><tr>
<th>#</th><th>File</th><th>Line</th><th>Problem</th></tr>
<tr bgcolor="lightgrey">
<td align="center">1</td>
<td width="*%">/root/github-repos/gradle-simple/src/main/java/Hello.java</td>
<td align="center" width="5%">12</td>
<td width="*"><a href="https://pmd.github.io/pmd-6.8.0/pmd_rules_java_errorprone.html#beanmembersshouldseria
lize">Found non-transient, non-static member. Please mark as transient or provide accessors.</a></td>
</tr>
<tr>
<td align="center">2</td>
<td width="*%">/root/github-repos/gradle-simple/src/main/java/Hello.java</td>
<td align="center" width="5%">15</td>
<td width="*"><a href="https://pmd.github.io/pmd-6.8.0/pmd_rules_java_errorprone.html#dataflowanomalyanalysi
s">Found 'DU'-anomaly for variable 'i' (lines '15'-'16').</a></td>
</tr>
</table></body></html>
root@attackdefense:~/github-repos/gradle-simple#
```

Issues Detected

- Found Non-transient, non-static member (Member variables should be marked as transient)
  Read more:
  https://pmd.github.io/latest/pmd_rules_java_errorprone.html#beanmembersshouldserialize
- Found ‘DU’ -anomaly (recently defined variable but never used)
  Read more:
  https://pmd.github.io/latest/pmd_rules_java_errorprone.html#dataflowanomalyanalysis

Learnings

Perform automated code review using PMD tool.