**STANDARD: Potentially Unsafe Code - java.io.File**

Line: 20 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

This functionality acts as an entry point for external data and the code should be manually checked to ensure the data obtained is correctly validated and/or sanitised. Additionally, carefull checks/sanitisation should be applied in any situation where the user may be able to control or affect the filename.

import java.io.File;

**STANDARD: Potentially Unsafe Code - FileInputStream**

Line: 21 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

This function acts as an entry point for external data and the code should be manually checked to ensure the data obtained is correctly validated and/or sanitised. Additionally, carefull checks/sanitisation should be applied in any situation where the user may be able to control or affect the filename.

import java.io.FileInputStream;

**STANDARD: Potentially Unsafe Code - java.io.File**

Line: 21 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

This functionality acts as an entry point for external data and the code should be manually checked to ensure the data obtained is correctly validated and/or sanitised. Additionally, carefull checks/sanitisation should be applied in any situation where the user may be able to control or affect the filename.

import java.io.FileInputStream;

**STANDARD: Potentially Unsafe Code - java.io.File**

Line: 22 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

This functionality acts as an entry point for external data and the code should be manually checked to ensure the data obtained is correctly validated and/or sanitised. Additionally, carefull checks/sanitisation should be applied in any situation where the user may be able to control or affect the filename.

import java.io.FileOutputStream;

**STANDARD: Potentially Unsafe Code - java.io.FileOutputStream**

Line: 22 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

This functionality acts as an entry point for external data and the code should be manually checked to ensure the data obtained is correctly validated and/or sanitised. Additionally, carefull checks/sanitisation should be applied in any situation where the user may be able to control or affect the filename.

import java.io.FileOutputStream;

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 29 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class MavenWrapperDownloader {

**STANDARD: Potentially Unsafe Code - FileInputStream**

Line: 65 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

This function acts as an entry point for external data and the code should be manually checked to ensure the data obtained is correctly validated and/or sanitised. Additionally, carefull checks/sanitisation should be applied in any situation where the user may be able to control or affect the filename.

FileInputStream mavenWrapperPropertyFileInputStream = null;

**STANDARD: Potentially Unsafe Code - FileInputStream**

Line: 67 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

This function acts as an entry point for external data and the code should be manually checked to ensure the data obtained is correctly validated and/or sanitised. Additionally, carefull checks/sanitisation should be applied in any situation where the user may be able to control or affect the filename.

mavenWrapperPropertyFileInputStream = new FileInputStream(mavenWrapperPropertyFile);

**MEDIUM: Potentially Unsafe Code - Failure To Release Resources In All Cases**

Line: 108 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

There appears to be no 'finally' block to release resources if an exception occurs, potentially resulting in DoS conditions from excessive resource consumption.

**MEDIUM: Potentially Unsafe Code - FileStream Opened Without Exception Handling**

Line: 108 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

There appears to be no 'try' block to safely open the filestream, potentially resulting in server-side exceptions.

**MEDIUM: Potentially Unsafe Code - Failure To Release Resources In All Cases**

Line: 108 - C:\dev\java\spades\.mvn\wrapper\MavenWrapperDownloader.java

There appears to be no release of resources in the 'finally' block, potentially resulting in DoS conditions from excessive resource consumption.

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 24 - C:\dev\java\spades\src\main\java\com\spades\spades\Deck.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

for (int i = 0; i < RANKS.length; i++) {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 25 - C:\dev\java\spades\src\main\java\com\spades\spades\Deck.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

for (int j = 0; j < SUITS.length; j++) {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 26 - C:\dev\java\spades\src\main\java\com\spades\spades\Deck.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

card = RANKS[i] + SUITS[j];

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 14 - C:\dev\java\spades\src\main\java\com\spades\spades\SpadesApplication.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class SpadesApplication {

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 21 - C:\dev\java\spades\src\main\java\com\spades\spades\SpadesApplication.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class HttpsRedirectConf {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 44 - C:\dev\java\spades\src\main\java\com\spades\spades\SpadesRoundImpl.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

for(int i = 0; i < 13; i++)

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 306 - C:\dev\java\spades\src\main\java\com\spades\spades\SpadesRoundImpl.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

int leadValue = -1;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 308 - C:\dev\java\spades\src\main\java\com\spades\spades\SpadesRoundImpl.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

int followValue = -1;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 329 - C:\dev\java\spades\src\main\java\com\spades\spades\SpadesRoundImpl.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

int winner = -1;

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 23 - C:\dev\java\spades\src\main\java\com\spades\spades\config\SecurityConfiguration.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class SecurityConfiguration extends WebSecurityConfigurerAdapter {

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 36 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ActiveGameController.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class ActiveGameController {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 183 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ActiveGameController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

int bidAmount = -1;

**STANDARD: Potentially Unsafe Code - getParameter**

Line: 186 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ActiveGameController.java

This function extracts data directly from the HTTP request. Manually check the code to ensure the data obtained is correctly validated and/or sanitised.

bidAmount = Integer.parseInt(req.getParameter("bidAmount"));

**HIGH: Potentially Unsafe Code - Poor Input Validation**

Line: 186 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ActiveGameController.java

The application appears to use data contained in the HttpServletRequest without validation or sanitisation. No validator plug-ins were located in the application's XML files.

bidAmount = Integer.parseInt(req.getParameter("bidAmount"));

**STANDARD: Potentially Unsafe Code - getParameter**

Line: 223 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ActiveGameController.java

This function extracts data directly from the HTTP request. Manually check the code to ensure the data obtained is correctly validated and/or sanitised.

String card = req.getParameter("card");

**HIGH: Potentially Unsafe Code - Poor Input Validation**

Line: 223 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ActiveGameController.java

The application appears to use data contained in the HttpServletRequest without validation or sanitisation. No validator plug-ins were located in the application's XML files.

String card = req.getParameter("card");

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 24 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\CreateGameController.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class CreateGameController {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 58 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\CreateGameController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

response += "<a href=\"/secured/all/game/" + gameId + "\">Go to new game</a>\n";

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 86 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\CreateGameController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

LOGGER.info("game created with id =" + newId);

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 28 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\CreateUserController.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class CreateUserController {

**STANDARD: Potentially Unsafe Code - getParameter**

Line: 43 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\CreateUserController.java

This function extracts data directly from the HTTP request. Manually check the code to ensure the data obtained is correctly validated and/or sanitised.

String userName = hReq.getParameter("username");

**STANDARD: Potentially Unsafe Code - getParameter**

Line: 44 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\CreateUserController.java

This function extracts data directly from the HTTP request. Manually check the code to ensure the data obtained is correctly validated and/or sanitised.

String lastName = hReq.getParameter("lastname");

**STANDARD: Potentially Unsafe Code - getParameter**

Line: 45 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\CreateUserController.java

This function extracts data directly from the HTTP request. Manually check the code to ensure the data obtained is correctly validated and/or sanitised.

String email = hReq.getParameter("email");

**STANDARD: Potentially Unsafe Code - getParameter**

Line: 46 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\CreateUserController.java

This function extracts data directly from the HTTP request. Manually check the code to ensure the data obtained is correctly validated and/or sanitised.

String password = hReq.getParameter("password");

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 12 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\GeneralErrorController.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class GeneralErrorController implements ErrorController {

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 19 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\HelloResource.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class HelloResource {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 55 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\HelloResource.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

if (getUserOpenGames(playerID1) != -1) {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 57 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\HelloResource.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

buttonOrLink = "<a href=\"/secured/all/game/" + gameId + "\">Go to Existing Open Game</a>\n";

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 26 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\JoinGameController.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class JoinGameController {

**STANDARD: Potentially Unsafe Code - getParameter**

Line: 46 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\JoinGameController.java

This function extracts data directly from the HTTP request. Manually check the code to ensure the data obtained is correctly validated and/or sanitised.

int gameId = Integer.parseInt(req.getParameter("game\_id"));

**HIGH: Potentially Unsafe Code - Poor Input Validation**

Line: 46 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\JoinGameController.java

The application appears to use data contained in the HttpServletRequest without validation or sanitisation. No validator plug-ins were located in the application's XML files.

int gameId = Integer.parseInt(req.getParameter("game\_id"));

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 58 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\JoinGameController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

String gameURL = "/secured/all/game/" + gameId;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 89 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\JoinGameController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

LOGGER.debug("Joined game id = " + gameId);

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 17 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\RetrieveOpenGamesController.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class RetrieveOpenGamesController {

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 26 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewEndedGamesController.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class ViewEndedGamesController {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 101 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewEndedGamesController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

String specificGameUrl = "/secured/all/viewendedgames/" + idGame;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 103 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewEndedGamesController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

listResponse.append("<a href=\"" + specificGameUrl + "\">View Spades Game ID # " + idGame + "</a><br/>");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 134 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewEndedGamesController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

for(int i = 0; i < players.size(); i++)

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 137 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewEndedGamesController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

result.append("<p>Player #" + (i+1) + ": " + u.getName());

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 172 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewEndedGamesController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

result.append("<p> Move " + moveCount + ": ");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 175 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewEndedGamesController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

moveCount++;

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 18 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewWinLossStatsController.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class ViewWinLossStatsController {

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 72 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewWinLossStatsController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

wins += 1;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 76 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewWinLossStatsController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

losses += 1;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 83 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewWinLossStatsController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

personalResponse.append("<p> Win/Loss: " + wins + "/" + losses + "</p>");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 87 - C:\dev\java\spades\src\main\java\com\spades\spades\resources\ViewWinLossStatsController.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

othersResponse.append("<p>Statistics for " + u.getName() + " (win/loss): " + wins + "/" + losses + "</p>");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 161 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

roundnum = temp.getRoundNumber() + 1;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 439 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

int actual1 = r.getPlayer1Actual() + 1;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 445 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

int actual2 = r.getPlayer2Actual() + 1;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 491 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

player1Points += calculatePoints(roundBid1, roundActual1);

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 492 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

player1Bags += calculateBags(roundBid1, roundActual1);

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 497 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

player2Points += calculatePoints(roundBid2, roundActual2);

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 498 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

player2Bags += calculateBags(roundBid2, roundActual2);

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 502 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

player1Points -= (player1Bags / 10) \* 100;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 503 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

player2Points -= (player2Bags / 10) \* 100;

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 678 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

for(int i = 0; i < players.size(); i++)

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 691 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

for(int i = 0; i < players.size(); i++)

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 717 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

for(int i = 0; i < players.size(); i++)

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 731 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

totalScore.set(i, accumulatedPoints + points);

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 733 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

totalBags.set(i, accumulatedBags + bags);

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 736 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

tableResult.append("<td>" + roundBid.get(i) + "</td>");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 737 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

tableResult.append("<td>" + roundActual.get(i) + "</td>");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 738 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

tableResult.append("<td>" + points + "</td>");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 739 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

tableResult.append("<td>" + bags + "</td>");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 748 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

for(int i = 0; i < players.size(); i++)

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 752 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

int finalPoints = rawPoints - ((numBags / 10) \* 100);

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 755 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

aggregateResults.append("<p>" + playerString + " Raw Score: " + rawPoints + ",");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 756 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

aggregateResults.append(" Total Bags: " + numBags + "<br/>\n");

**LOW: Potentially Unsafe Code - Operation on Primitive Data Type**

Line: 757 - C:\dev\java\spades\src\main\java\com\spades\spades\service\SpadesGameService.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

aggregateResults.append(playerString + " Adjusted Score (subtract bags): " + finalPoints + "\n");

**POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final**

Line: 10 - C:\dev\java\spades\src\test\java\com\spades\spades\SpadesApplicationTests.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class SpadesApplicationTests {