

# Homework 1

**Student:** Patrick Walsh

**School:** University of Maryland Global Campus

**Course:** SDEV 425 6980

**Professor:** Dr. Nicholas Duchon

## Part I: Environment Set-up and Simple Hello, World

Simple Hello World program:

```
-----< com.mycompany:SDEV425-week2 >-----
[-] Building SDEV425-week2 1.0-SNAPSHOT
-----[ jar ]-----

[-] --- exec-maven-plugin:3.0.0:exec (default-cli) @ SDEV425-week2 ---
*****
Hello World!!
SDEV 425, the current date and time is:
2021/06/17 18:31:02
*****

BUILD SUCCESS

Total time: 1.881 s
Finished at: 2021-06-17T18:31:02-04:00
-----
```

## Part II: Fix security issues in a simple Java application that uses command line arguments.

The program in its current state compiles and runs successfully, reading the .txt file and printing out its contents:

```
[-] --- exec-maven-plugin:3.0.0:exec (default-cli) @ SDEV425_1 ---
Reading file EmailAddresses-UNSAFE-VERSION.txt...

Email Addresses:
john@umgc.edu
fred@umgc.edu
susan@umgc.edu
donna@umgc.edu
javier@umgc.edu
jessie@umgc.edu
laura@umgc.edu
tina@umgc.edu
todd@umgc.edu
ed@umgc.edu

BUILD SUCCESS

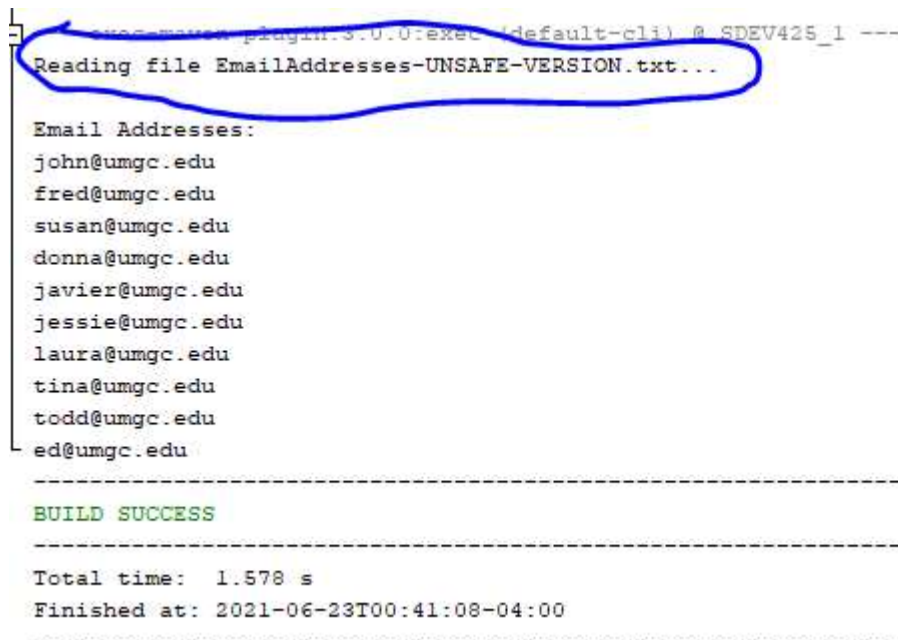
Total time: 1.578 s
Finished at: 2021-06-23T00:41:08-04:00
-----
```

However, the program violates security rule [IDS50-J. Use conservative file naming conventions](#) under Input Validation and Data Sanitization. Namely, the program takes in unsanitized input from the user where it is expecting a file name. See lines 30 and 50:

```
29 | // Read the filename from the command line argument
30 | String filename = args[0];

49 | try {
50 |     inputStream = new BufferedReader(new FileReader(filename));
```

Characters such as leading dashes, control characters, and spaces can cause security issues when Java reads a file name since these characters can be exploited by attackers to make the program behave in unexpected ways (Svoboda, 2016). Since the program is not checking the file name input, an unsafe file name can be used:



```
plugin.3.0.0:exec (default-cli) @ SDEV425_1 ---
Reading file EmailAddresses-UNSAFE-VERSION.txt...

Email Addresses:
john@umgc.edu
fred@umgc.edu
susan@umgc.edu
donna@umgc.edu
javier@umgc.edu
jessie@umgc.edu
laura@umgc.edu
tina@umgc.edu
todd@umgc.edu
ed@umgc.edu

-----
BUILD SUCCESS
-----

Total time: 1.578 s
Finished at: 2021-06-23T00:41:08-04:00
-----
```

To fix this security issue, the input should be sanitized through a Regular Expression name check. The code snippet below takes in the file name input and sanitizes the input to make sure only uppercase letters, lowercase letters, numbers, periods, and underscores are being used for the file name:

```

28 // ***** NEW CODE *****
29 // checks to make sure filename only contains uppercase letters,
30 // lowercase letters, numbers, periods, and underscores
31 System.out.println("Reading file " + filename + "...\\n");
32 Pattern pattern = Pattern.compile("[^A-Za-z0-9._]");
33 Matcher matcher = pattern.matcher(filename);
34 if (matcher.find()) {
35 // File name contains bad chars; handle error
36 System.out.println("Invalid filename! Contains characters other "
37 + "than uppercase letters, lowercase letters, numbers, "
38 + "periods, or underscores");
39 System.out.println("\\nEXITING PROGRAM...");
40 System.exit(0);
41 }
42 // ***** END OF NEW CODE *****

```

Now when the same input is used, the program catches the security issue and exits the program before the file is read:

```

] Building SDEV425_1 1.0-SNAPSHOT
-----[ jar ]-----

] --- exec-maven-plugin:3.0.0:exec (default-cli) @ SDEV425_1 ---
Reading file EmailAddresses-UNSAFE-VERSION.txt...

Invalid filename! Contains characters other than uppercase letters, lowercase letters, numbers, periods, or underscores

EXITING PROGRAM...

BUILD SUCCESS

Total time: 1.697 s
Finished at: 2021-06-23T00:53:25-04:00
-----

```

Another invalid input is attempted and the program catches this security issue as well:

```

-----< com.mycompany:SDEV425_1 >-----
] Building SDEV425_1 1.0-SNAPSHOT
-----[ jar ]-----

] --- exec-maven-plugin:3.0.0:exec (default-cli) @ SDEV425_1 ---
Reading file EmailAddressesUN$AFEV3R$IION.txt...

Invalid filename! Contains characters other than uppercase letters, lowercase letters, numbers, periods, or underscores

EXITING PROGRAM...

BUILD SUCCESS

Total time: 1.653 s
Finished at: 2021-06-23T00:49:47-04:00
-----

```

Now a valid input is tried, and the program successfully compiles and runs:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ SDEV425_1 ---  
Reading file EmailAddresses.txt...
```

Email Addresses:

```
john@umgc.edu  
fred@umgc.edu  
susan@umgc.edu  
donna@umgc.edu  
javier@umgc.edu  
jessie@umgc.edu  
laura@umgc.edu  
tina@umgc.edu  
todd@umgc.edu  
ed@umgc.edu
```

-----  
**BUILD SUCCESS**  
-----

Total time: 1.566 s

Finished at: 2021-06-23T00:55:33-04:00  
-----

## References

Svoboda, D. (2016, October 5). *IDS50-J. Use conservative file naming conventions - SEI CERT Oracle Coding Standard for Java - Confluence*. Carnegie Mellon University Software Engineering Institute.  
<https://wiki.sei.cmu.edu/confluence/display/java/IDS50-J.+Use+conservative+file+naming+conventions>