Security Injections @Towson

Welcome, Marilyn Soyars!

Security Injections, Java CS1 - Buffer Overflow



Laboratory Assignment

PART-1

STEP 1: Type(do not copy and paste) Program 1 into a program and compile. Run the program

Program 1:

```
import java.util.Scanner;

public class Overflow2 {
  public static void main(String[] args) {

    Scanner scan = new Scanner(System.in);
    int[] vals = new int[10];

    System.out.println("How many values should be stored in the array? ");
    int count = scan.nextInt();

    for (int i = 0; i < count; i++ ) {
        vals[i] = count-i;
    }

    System.out.println("Which value do you wish to retrieve? ");
    int which = scan.nextInt();

    System.out.println("Your value is " +vals[which]);
    }
}</pre>
```

Question 1

Describe the results of above program run in step 1?

If you type a number less than or equal to 10 forthe first prompt, and a number between 0 and 9 for the second prompt, the program will execute.

Question 2

What happens if you type "3" for the first prompt?

Program executes without error

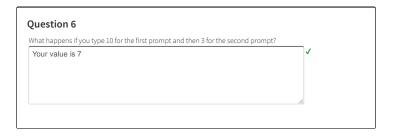
Question 3

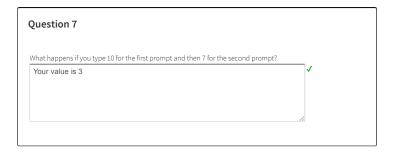
What happens if you type "7" for the first prompt?

Program executes without error

Question 4 What happens if you type "12" for the first prompt? Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10 at BufferOverflow.main(BufferOverflow.java:13)

Question 5 What happens if you type "20" for the first prompt? Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10 at BufferOverflow.main(BufferOverflow.java:13)





```
Question 8

What happens if you type 10 for the first prompt and then 12 for the second prompt?

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 12
at BufferOverflow.main(BufferOverflow.java:19)
```

Question 9 What happens if you type 10 for the first prompt and then 20 for the second prompt? Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 20 at BufferOverflow.main(BufferOverflow.java:19)

 $\ensuremath{\mathsf{STEP\,2:}}$ Complete the security checklist for the program created in Step 1

► Click to see how a checklist works

Question 10

```
import java.util.Scanner;
public class Overflow2 {
  public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    int[] vals = new int[10];

    System.out.println("How many values should be stored in the array? ");
    int count = scan.nextInt();

    for (int i = 0; i < count; i++) { //0 <= i < 10
        vals[i] = count - i; // 0 <= i < 10
    }

    System.out.println("Which value do you wish to retrieve? ");
    int which = scan.nextInt();

    System.out.println("Your value is " + vals[which]);
}
</pre>
```

Vulnerability: Integer Errors Course: CS1	
Check each line of code	Completed
1. Finding Arrays:	
1.1 Click each array declaration	>
1.2 For each array, click all subsequent references	✓
2. Index Variables - legal range for an array of size n is 0 <= i < n	
2.1 For each array access that uses a variable as an index, fill in the legal range next to it.	✓
2.2 For each index variable from the array access in 2.1, click all occurrences of that variable.	✓
 Click any assignments, inputs or operations that may modify these index variables. 	✓ ✓
2.4. Click any array that is indexed by a clicked index variable.	✓ ✓
3. Loops that modify index variables	
3.1 Find loops that modify variables used to index arrays. For any index that occurs as part of a loop conditional, click the loop limit. For example, if i < max is the conditional in a for loop, click max	₹ ✓
3.2. Fill in the legal range of the array index next to the loop limit as	₩ ✓
you did in step 2.1. Click if the loop limit could exceed the legal range of the array index. Watch out for loops that go until $i \leftarrow max$, as the largest valid index is $max - 1$	
3.3 If the upper or lower loop limit is a variable, it must be checked just as indices are checked in Step 2	
Highlighted areas indicate vulnerabilities!	

Question 11

List places where the bounds checking should occur?

count may be greater than 9 or less than zero which may be greater than 9 or less than zero

Question 12

 $\underbrace{ \text{Provide example inputs that might cause array index out of bounds exception?} }$

If input values are greater than or equal to 10 will result in buffer overflow

STEP 4: Rewrite the above program to include the appropriate bounds checking.

PART-2

 $\textbf{STEP 1:} \ \ \mathsf{Create} \ \mathsf{the} \ \mathsf{following} \ \mathsf{java} \ \mathsf{file} \ \mathsf{in} \ \mathsf{a} \ \mathsf{java} \ \mathsf{complier} \mathsf{:}$

```
import java.io.*; // for File
import java.util.*; // for Scanner
public class ReadTemps
{
```

```
public static void main (String[] args) throws FileNotFoundException
{
    Scanner inFile = new Scanner(new File("temps.txt")); //open
    double[] temps = new double[10];
    int numTemps = 0;

    while (inFile.hasNextDouble())
    {
        temps[numTemps] = inFile.nextDouble();
        numTemps++;
    }
    System.out.println(numTemps + " temperatures were read.");
}
```

STEP 2: Create the following temps.txt file(Make sure this file is in the same folder where the above java file is stored):

```
30.0 30.1 30.2 30.3 30.4 30.5 30.6 20.7 30.8 30.9
```

STEP 3: Compile and run the program. It should list 10 temperatures as being read.

STEP 4: Change temps.txt to the following:

```
30.0 30.1 30.2 30.3 30.4 30.5 30.6 20.7 30.8 30.9 31.0 31.2
```

STEP 5: Compile and run the program.

Question 13

What happended after compiling and running the program in step 5 and why?

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10 at ReadTemps.main(ReadTemps.java:13)

because the maximum array is 10

STEP 6: Modify the program to fix the problem. DO NOT change the size of the array! Hint: Add a check to the loop.

Go To Next Section



This project is supported by the National Science Foundation under grants DUE-1241738 and DUE-0817267. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Powered by a modified version of Class2Go