Lab #7: Android RE

CYBR 570 : Reverse Engineering
Spring 2025

Submitted By: Lilly Carlascio
Submission Date: 5/21/25

Setup:

- I first downloaded the dice-rolling app on my fire tablet to test for functionality. I then uninstalled the app so I could modify it and redownload the modified application.

- I used apk tool to decompile the apk into its own firectory using this command

```
java -jar ~/Downloads/apktool.jar d org.secuso.privacyfriendlydicer_12.apk
-o dice-mod
```

Step 1: Finding the location where a random number generator is used to "roll the dice".

- To first find the location where the number of the dice was being chosen, I had to google what java method randomly selects a number from a given set of numbers. (Ex: for 6 numbers, either 1, 2, 3, 4, 5, 6 can be randomly chosen). The search showed me that the function nextInt() returns a random integer from a given set of numbers.

- From the command line I ran this command:

```
grep -R "nextInt" dice-mod/smali/
```

  which gave me the output seen in figure 1. There was a lot more from various kotlin packages, but the file I chose to focus on was "dice-mod/smali/org/secuso/privacyfriendlydicer/dicer/Dicer.smali" since the file was called Dicer.smali, which proved to be important.

Step 2: Change parameter to the function call (max-1).

- Once inside that file, I searched for "nextInt" and found it in a function called rollDice(II) which was where I would be doing the modification of code. Seeing nextInt() in this function showed me that it took in the value of p2, which appeared to be the max value passed in and then the result is moved into v2, another register.

- As for modifications, before the nextInt method is called, I first updated the .locals in the first line of the function to be 6 instead of 3 since I would be needing to use additional registers. Then I added lines of code that calculate a new max value to be passed into the nextInt() method as seen here:

```
const/4 v3, -0x1
add-int v3, p2, v3
invoke-virtual {v2, v3}, Ljava/security/SecureRandom;->nextInt(I)I
```

This ensures that the max value that can be rolled will always be 1 less than the max value possible. v3 is the new max value being passed into the nextInt() method and then that result is moved into v2 in the next line of code.

Step 3: Add smali code to print the newly decremented parameter to logcat.

- To add a print statement showing the new max value to the command line, I used the helpful hint provided by the lab document to run this command from the dice-mod directory:

```
grep -r "Landroid/util/Log"
```

which gave me a lot of files that print logcat messages to the command line. I ended up chosing a random one to look at (dice-mod-test/smali/androidx/appcompat/widget/MenuPopup Window.smali) and found these lines of code:

```
const-string v0, "MenuPopupWindow"

const-string v1, "Could not find method setTouchModal() on PopupWindow.
Oh well."

invoke-static {v0, v1}, Landroid/util/Log;->i(Ljava/lang/String;Ljava/
lang/String;)I
```

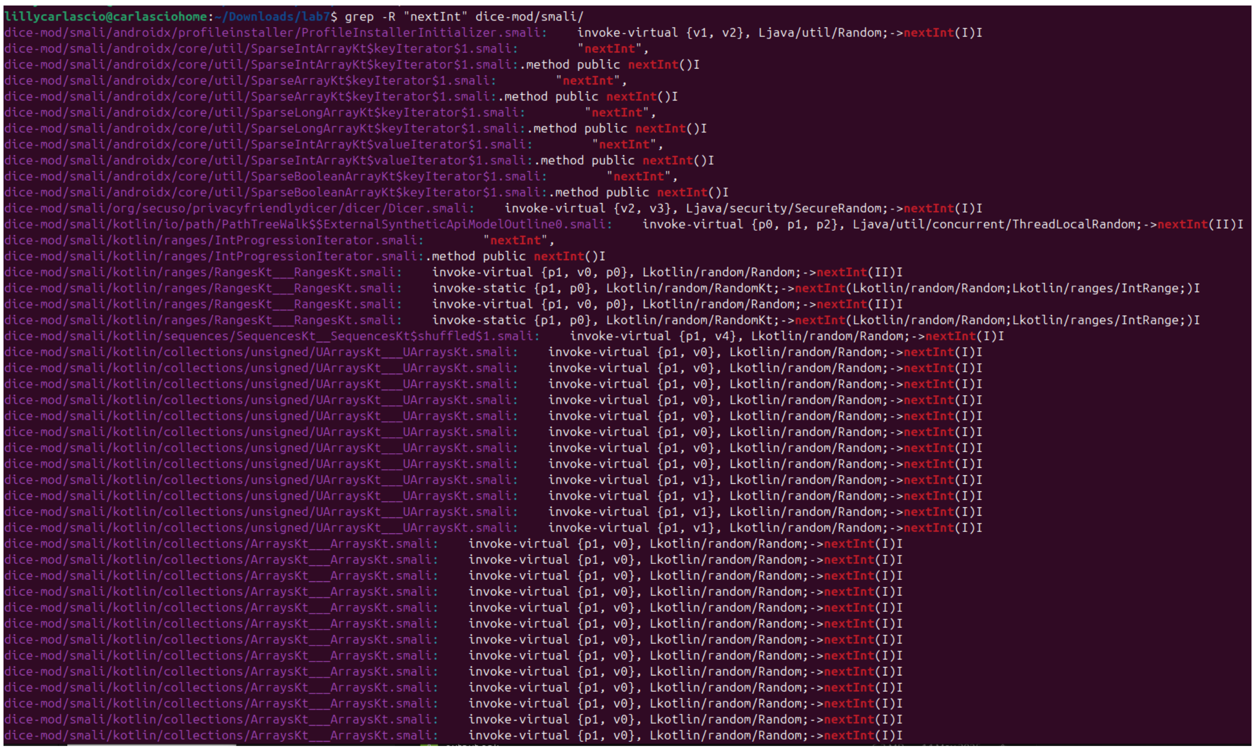
Which showed me that the first string (MenuPopupWindow) is being loaded into v0, then the second string (Could not find...) is being loaded into v0. And finally the logging method is being called with an 'i' tag (info) of MenuPopupWindow and the message being the longer string that will get printed to the console.

- This leads me to the code I added for logging the new max value everytime a set of dice is rolled. This is the code I added:

```
const-string v4, "Max Value that can be rolled: "
invoke-static {v3}, Ljava/lang/String;->valueOf(I)Ljava/lang/String;
move-result-object v5
invoke-static {v4, v5}, Landroid/util/Log;->d(Ljava/lang/String;Ljava
/lang/String;)I
```

This code is taking a new string (Max Value...) and the value of v3 which was the new max and it first must convert the value of v3 into a string so it can be printed with the message, and then it is stored into a new register v. Those values are then printed by logcat in the final line of code. The 'd' flag (debug) is used in this case since we are printing for debugging purposes and it is lower priority than the 'i' flag used before in the other file.

- An example output of the logcat loggin messages can be seen in figure 2 with examples of 9, 6 and 10 dice being rolled. But as you can see, the max being shown are 8, 5, and 9.



```
lillycarlascio@carlasciohome:~/Downloads/lab7$ grep -R "nextInt" dice-mod/smali/
dice-mod/smali/androidx/profileinstaller/ProfileInstallerInitializer.smali:    invoke-virtual {v1, v2}, Ljava/util/Random;->nextInt(I)I
dice-mod/smali/androidx/core/util/SparseIntArrayKt$keyIterator$1.smali:        "nextInt",
dice-mod/smali/androidx/core/util/SparseIntArrayKt$keyIterator$1.smali:.method public nextInt()I
dice-mod/smali/androidx/core/util/SparseArrayKt$keyIterator$1.smali:        "nextInt",
dice-mod/smali/androidx/core/util/SparseArrayKt$keyIterator$1.smali:.method public nextInt()I
dice-mod/smali/androidx/core/util/SparseLongArrayKt$keyIterator$1.smali:        "nextInt",
dice-mod/smali/androidx/core/util/SparseLongArrayKt$keyIterator$1.smali:.method public nextInt()I
dice-mod/smali/androidx/core/util/SparseIntArrayKt$valueIterator$1.smali:        "nextInt",
dice-mod/smali/androidx/core/util/SparseIntArrayKt$valueIterator$1.smali:.method public nextInt()I
dice-mod/smali/androidx/core/util/SparseBooleanArrayKt$keyIterator$1.smali:        "nextInt",
dice-mod/smali/androidx/core/util/SparseBooleanArrayKt$keyIterator$1.smali:.method public nextInt()I
dice-mod/smali/org/secuso/privacyfriendlydicer/dicer/Dicer.smali:    invoke-virtual {v2, v3}, Ljava/security/SecureRandom;->nextInt(I)I
dice-mod/smali/kotlin/io/path/PathTreeWalk$$ExternalSyntheticApiModelOutline0.smali:    invoke-virtual {p0, p1, p2}, Ljava/util/concurrent/ThreadLocalRandom;->nextInt(II)I
dice-mod/smali/kotlin/ranges/IntProgressionIterator.smali:        "nextInt",
dice-mod/smali/kotlin/ranges/IntProgressionIterator.smali:.method public nextInt()I
dice-mod/smali/kotlin/ranges/RangesKt___RangesKt.smali:    invoke-virtual {p1, v0, p0}, Lkotlin/random/Random;->nextInt(II)I
dice-mod/smali/kotlin/ranges/RangesKt___RangesKt.smali:    invoke-static {p1, p0}, Lkotlin/random/RandomKt;->nextInt(Lkotlin/random/Random;Lkotlin/ranges/IntRange;)I
dice-mod/smali/kotlin/ranges/RangesKt___RangesKt.smali:    invoke-virtual {p1, v0, p0}, Lkotlin/random/Random;->nextInt(II)I
dice-mod/smali/kotlin/ranges/RangesKt___RangesKt.smali:    invoke-static {p1, p0}, Lkotlin/random/RandomKt;->nextInt(Lkotlin/random/Random;Lkotlin/ranges/IntRange;)I
dice-mod/smali/kotlin/sequences/SequencesKt__SequencesKt$shuffled$1.smali:    invoke-virtual {p1, v4}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v1}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v1}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v1}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/unsigned/UArraysKt___UArraysKt.smali:    invoke-virtual {p1, v1}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
dice-mod/smali/kotlin/collections/ArraysKt___ArraysKt.smali:    invoke-virtual {p1, v0}, Lkotlin/random/Random;->nextInt(I)I
```

Figure 1: Output from grep command searching for nextInt() java function.

```
lillycarlascio@carlasciohome:~/Downloads/lab7$ adb logcat | grep Max
05-20 10:13:11.921 30659 30659 D Max Value that can be rolled: : 8
05-20 10:13:11.921 30659 30659 D Max Value that can be rolled: : 8
05-20 10:13:12.652 30659 30659 D Max Value that can be rolled: : 8
05-20 10:13:12.652 30659 30659 D Max Value that can be rolled: : 8
05-20 10:13:13.427 30659 30659 D Max Value that can be rolled: : 8
05-20 10:13:13.427 30659 30659 D Max Value that can be rolled: : 8
05-20 10:13:54.748 31548 31548 D Max Value that can be rolled: : 5
05-20 10:13:54.749 31548 31548 D Max Value that can be rolled: : 5
05-20 10:13:55.172 31548 31548 D Max Value that can be rolled: : 5
05-20 10:13:55.172 31548 31548 D Max Value that can be rolled: : 5
05-20 10:13:55.554 31548 31548 D Max Value that can be rolled: : 5
05-20 10:13:55.555 31548 31548 D Max Value that can be rolled: : 5
05-20 10:13:56.304 31548 31548 D Max Value that can be rolled: : 5
05-20 10:13:56.304 31548 31548 D Max Value that can be rolled: : 5
05-20 10:13:58.270 31548 31548 D Max Value that can be rolled: : 9
05-20 10:13:58.270 31548 31548 D Max Value that can be rolled: : 9
05-20 10:13:58.618 31548 31548 D Max Value that can be rolled: : 9
05-20 10:13:58.618 31548 31548 D Max Value that can be rolled: : 9
05-20 10:13:58.971 31548 31548 D Max Value that can be rolled: : 9
05-20 10:13:58.971 31548 31548 D Max Value that can be rolled: : 9
05-20 10:13:59.334 31548 31548 D Max Value that can be rolled: : 9
05-20 10:13:59.334 31548 31548 D Max Value that can be rolled: : 9
```

Figure 2: Output from logcat debugger.