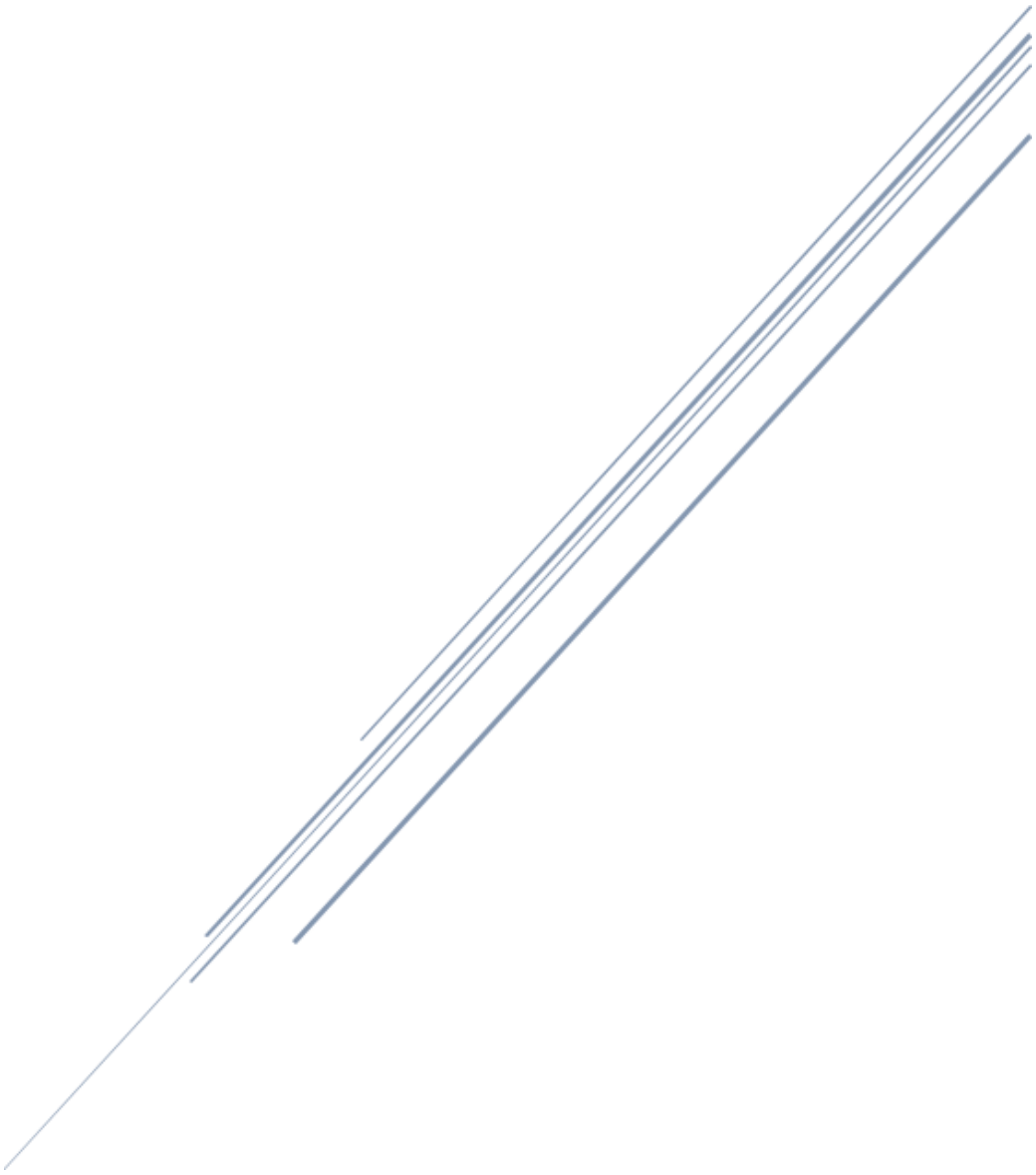
Bowshock

Space Race (Deloitte CTF)



By - p4cm4n

Github - (xxxxx)
TryHackMe - (xxxxx)

Hello guys! Today we are going to solve the Bowshock CTF challenge presented by Deloitte. It is a reverse engineering challenge where we need to check the source code for flaws and find the flag. To access this CTF you need to register for Deloitte CTF on https://hackyholidays.io/ (expired).

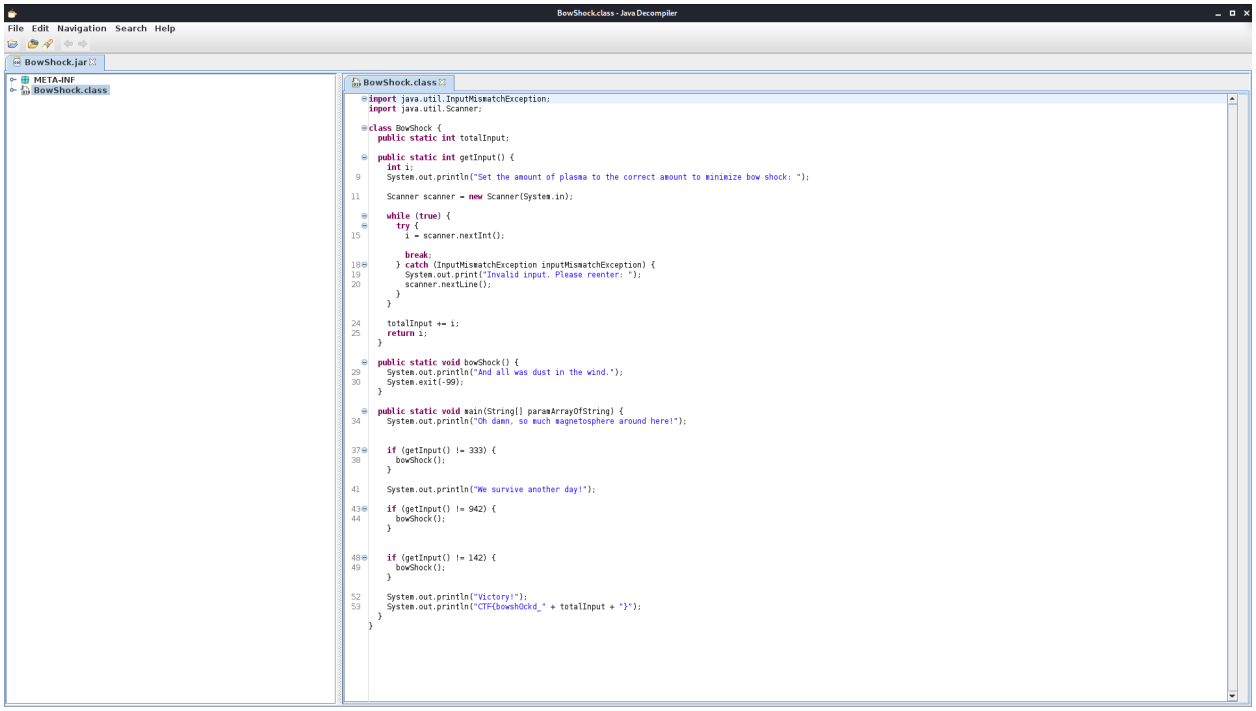
What we know!

- bowshock.jar file (source code)

.jar extension is for java language. You need some understanding of the Java language.

Opening the Jar file

I used jd-gui to open the .jar file. Jd-gui is a java debugger GUI to open java source code files. In kali linux jd-gui is in the repository, you can download it using apt command.



Jd-gui interface

```java
import java.util.InputMismatchException;
import java.util.Scanner;

class BowShock {
  public static int totalInput;

  public static int getInput() {
    int i;
    System.out.println("Set the amount of plasma to the correct amount to minimize bow shock: ");

    Scanner scanner = new Scanner(System.in);

    while (true) {
      try {
        i = scanner.nextInt();

        break;
      } catch (InputMismatchException inputMismatchException) {
        System.out.print("Invalid input. Please reenter: ");
        scanner.nextLine();
      }
    }

    totalInput += i;
    return i;
  }

  public static void bowShock() {
    System.out.println("And all was dust in the wind.");
    System.exit(-99);
  }

  public static void main(String[] paramArrayOfString) {
    System.out.println("Oh damn, so much magnetosphere around here!");


    if (getInput() != 333) {
      bowShock();
    }

    System.out.println("We survive another day!");

    if (getInput() != 942) {
      bowShock();
    }


    if (getInput() != 142) {
      bowShock();
    }

    System.out.println("Victory!");
    System.out.println("CTF{bowsh0ckd_" + totalInput + "}");
  }
}
```

Source code (bowshock.jar)

Analysing the code

After analysing the code I found some input values in the main function which were tested before triggering the function bowshock() which will exit the program.

```
  public static void main(String[] paramArrayOfString) {
34  System.out.println("Oh damn, so much magnetosphere around here!");


37  if (getInput() != 333) {
38    bowShock();
    }

41  System.out.println("We survive another day!");

43  if (getInput() != 942) {
44    bowShock();
    }


48  if (getInput() != 142) {
49    bowShock();
    }

52  System.out.println("Victory!");
53  System.out.println("CTF{bowshOckd_" + totalInput + "}");
  }
}
```

main() function

As you can see in the above screenshot the function main() checks the input against some predefined values which are 333, 942, and 142 respectively. If the user inputs a value other than the defined values function bowshock() is called.

```
  public static void bowShock() {
29  System.out.println("And all was dust in the wind.");
30  System.exit(-99);
  }
```

bowshock() function

From the above screenshot we can see that function bowshock() terminates the program.

Now by analysing the source code we know the values in the correct order that are 333, 942 , and 142 respectively.

Getting the Flag

To run the .jar file you need to install java JDK. In kali linux java JDK is already installed and if not then install it using apt command.

```
java -jar BowShock.jar
```

```
┌──(blinky㉿PacMan)-[~/bowshock]
└─$ java -jar BowShock.jar
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Oh damn, so much magnetosphere around here!
Set the amount of plasma to the correct amount to minimize bow shock:
333
We survive another day!
Set the amount of plasma to the correct amount to minimize bow shock:
942
Set the amount of plasma to the correct amount to minimize bow shock:
142
Victory!
CTF{bowsh0ckd_1417}
```

Enter the values in the correct order and we get the flag.

Flag: CTF{bowsh0ckd_1417}