SWE619 Assignment-12

-Amish Papneja

-Avinash Arunachalam A Murugappan

- Rushil Nandan Dubey

Question 3: ***For most of the semester, we have focused on design considerations for constructing software that does something we want it to do. For this last assignment, I would like students to appreciate just how vulnerable software is to malicious parties intent on attacking their software. Students who find this assignment amusing might wish to take CS 468: Secure Programming and Systems.***

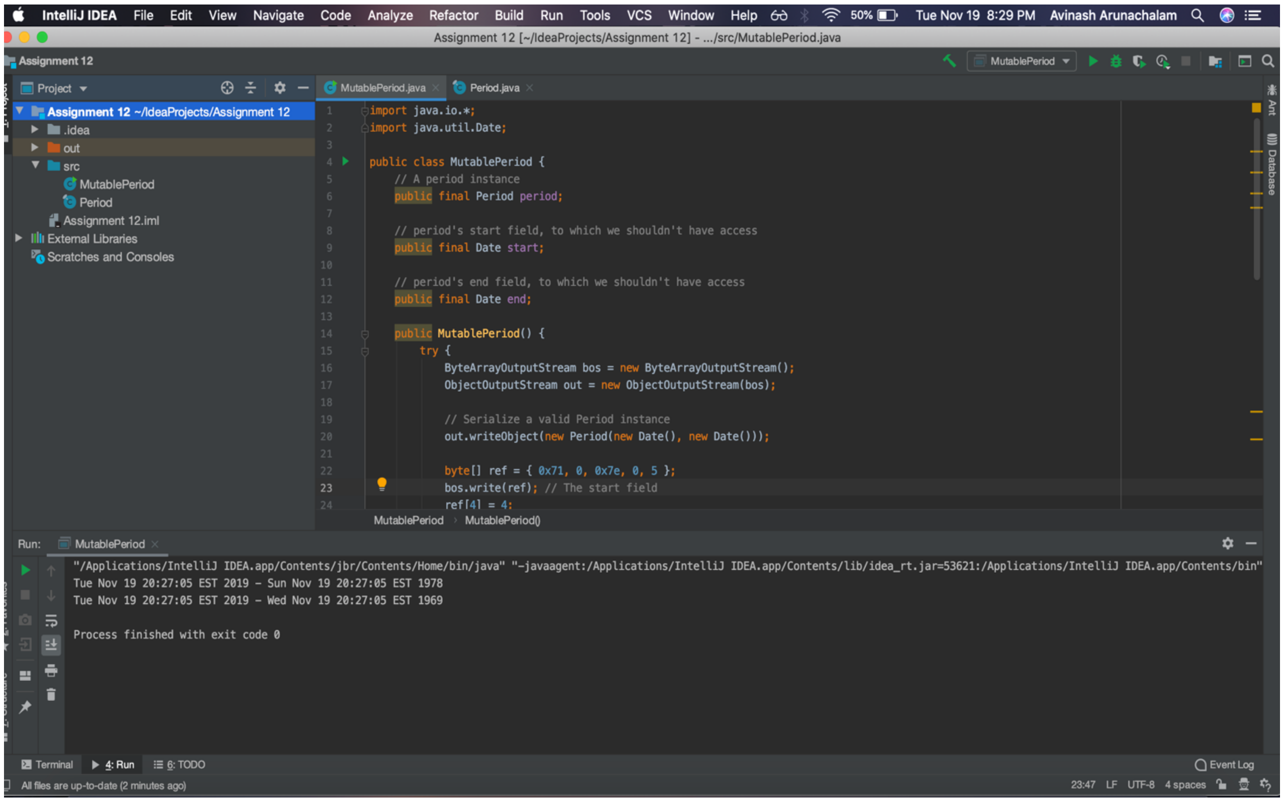
There are two attacks documented in Bloch's Item 88: Write `readObject()` methods defensively. One is called `BogusPeriod`, and the other is called `MutablePeriod`. Implement either (your choice) of these attacks (basically involves typing in code from Bloch) and verify that the attack takes place.

Answer:

We have choosen ***`MutablePeriod`*** and implemented it for this assignment.

From Bloch's Item 88, we notice that by setting pEnd equal to the MutablePeriod's end date, it's possible to modify pEnd so that it changes the MutablePeriod's date. This would be an issue if an attacker passes on an instance of the MutablePeriod class to a class that depends, for security, on a period's immutability. Serializable and deserializable classes, which can be passed around, need to have defensive copying instead.

We have attached a screenshot below to show that we have verified the attack has taken place, by modifying the pEnd and setting the year we want: which is 78,69.


IntelliJ IDEA File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Tue Nov 19 8:29 PM Avinash Arunachalam
Assignment 12 [~/IdeaProjects/Assignment 12] - .../src/MutablePeriod.java
Project
Assignment 12 ~/IdeaProjects/Assignment 12
.idea
out
src
MutablePeriod
Period
Assignment 12.iml
External Libraries
Scratches and Consoles
MutablePeriod.java
Period.java
Run: MutablePeriod
Terminal
4: Run
6: TODO
Event Log
All files are up-to-date (2 minutes ago)

```java
import java.io.*;
import java.util.Date;

public class MutablePeriod {
    // A period instance
    public final Period period;

    // period's start field, to which we shouldn't have access
    public final Date start;

    // period's end field, to which we shouldn't have access
    public final Date end;

    public MutablePeriod() {
        try {
            ByteArrayOutputStream bos = new ByteArrayOutputStream();
            ObjectOutputStream out = new ObjectOutputStream(bos);

            // Serialize a valid Period instance
            out.writeObject(new Period(new Date(), new Date()));

            byte[] ref = { 0x71, 0, 0x7e, 0, 5 };
            bos.write(ref); // The start field
            ref[4] = 4;
```

```
"/Applications/IntelliJ IDEA.app/Contents/jbr/Contents/Home/bin/java" "-javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=53621:/Applications/IntelliJ IDEA.app/Contents/bin"
Tue Nov 19 20:27:05 EST 2019 - Sun Nov 19 20:27:05 EST 1978
Tue Nov 19 20:27:05 EST 2019 - Wed Nov 19 20:27:05 EST 1969

Process finished with exit code 0
```

Code: MutablePeriod.Java

```
import java.io.*;
import java.util.Date;

public class MutablePeriod {
   // A period instance
   public final Period period;

   // period's start field, to which we shouldn't have access
   public final Date start;

   // period's end field, to which we shouldn't have access
   public final Date end;

   public MutablePeriod() {
      try {
         ByteArrayOutputStream bos = new ByteArrayOutputStream();
         ObjectOutputStream out = new ObjectOutputStream(bos);

         // Serialize a valid Period instance
         out.writeObject(new Period(new Date(), new Date()));

         byte[] ref = { 0x71, 0, 0x7e, 0, 5 };
         bos.write(ref); // The start field
         ref[4] = 4;
         bos.write(ref); // The end field

         // Deserialize Period and "stolen" Date references
         ObjectInputStream in = new ObjectInputStream(new
ByteArrayInputStream(bos.toByteArray()));
         period = (Period) in.readObject();
         start = (Date) in.readObject();
         end = (Date) in.readObject();
      } catch (IOException | ClassNotFoundException e) {
         throw new AssertionError(e);
      }
   }

   public static void main(String[] args) {
      MutablePeriod mp = new MutablePeriod();
      Period p = mp.period;
      Date pEnd = mp.end;
```

```
        // Let's turn back the clock
        pEnd.setYear(78);
        System.out.println(p);

        // Bring back the 60s!
        pEnd.setYear(69);
        System.out.println(p);
    }
}
```

Period.Java:

```
import java.util.Date;
import java.io.Serializable;
// Immutable class that uses defensive copying
public final class Period implements Serializable{
    private final Date start;
    private final Date end;

    public Period(Date start, Date end) {
        this.start = new Date(start.getTime());
        this.end = new Date(end.getTime());
        if (this.start.compareTo(this.end) > 0) {
            throw new IllegalArgumentException(start + " after " + end);
        }
    }

    public Date start () {return new Date(start.getTime()); }
    public Date end () {return new Date(end.getTime()); }
    public String toString() {return start + " - " + end;}
}
```

Contributions:

-Amish Papneja

- Helped implement and discuss story for assignment 12
- Contributed to discussion regarding this assignment

-Avinash Arunachalam A Murugappan

- Helped implement and discuss story for assignment 12
- Contributed to discussion regarding this assignment

- Rushil Nandan Dubey

- Helped implement and discuss story for assignment 12
- Contributed to discussion regarding this assignment