Jackson-Databind - A Story of Blacklisting Java Deserialization Gadgets

Gal Goldshtein, 2018-29-03

Jackson is a popular library for parsing JSON documents in Java. Jackson-Databind is a module of the Jackson library that allows automatic transformation from JSON to Java objects and vice versa.

In June 2017, an unsafe deserialization vulnerability was discovered in the Jackson-Databind module (CVE-2017-7525) and was patched by creating a blacklist that doesn't accept certein dangerous Java classes.

The Jackson-Databind module developers created a function named checkIllegalTypes and the only class that was blacklisted was TemplatesImpl which is part of the com.sun.org.apache.xalan package.

```
protected void checkIllegalTypes(DeserializationContext ctxt, JavaType type,
        BeanDescription beanDesc)
    throws JsonMappingException
{
    // There are certain nasty classes that could cause problems, mostly
    // via default typing -- catch them here.
    Class<?> raw = type.getRawClass();
    String name = raw.getSimpleName();

    if ("TemplatesImpl".equals(name)) { // [databind#1599]
        if (raw.getName().startsWith("com.sun.org.apache.xalan")) {
            throw JsonMappingException.from(ctxt,
                    String.format("Illegal type (%s) to deserialize: prevented for security reasons",
                            name));
        }
    }
}
```

Figure 1: The checkIllegalTypes function as it was first written in the BeanDeserializerFactory.java file

An hour passed, and the blacklist was updated to contain more dangerous classes.

```
/**
 * Set of well-known "nasty classes", deserialization of which is considered dangerous
 * and should (and is) prevented by default.
 *
 * @since 2.8.9
 */
protected final static Set<String> DEFAULT_NO_DESER_CLASS_NAMES;
static {
    Set<String> s = new HashSet<>();
    // Courtesy of [https://github.com/kantega/notsoserial]:
    // (and wrt [databind#1599]
    s.add("org.apache.commons.collections.functors.InvokerTransformer");
    s.add("org.apache.commons.collections.functors.InstantiateTransformer");
    s.add("org.apache.commons.collections4.functors.InvokerTransformer");
    s.add("org.apache.commons.collections4.functors.InstantiateTransformer");
    s.add("org.codehaus.groovy.runtime.ConvertedClosure");
    s.add("org.codehaus.groovy.runtime.MethodClosure");
    s.add("org.springframework.beans.factory.ObjectFactory");
    s.add("com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl");
    DEFAULT_NO_DESER_CLASS_NAMES = Collections.unmodifiableSet(s);
}
```

Figure 2: More dangerous classes were added to the blacklist.

In December 2017 CVE-2017-17485 was allocated for this exact same vulnerability because it was found that the blacklist didn't contain dangerous classes from the Spring framework.

After a while another bypass on the blacklist is discovered, now being tracked as CVE-2017-15095.

This is probably not the last change being made on this blacklist because new dangerous classes and gadget chains (Several classes that when combined may lead to Remote Code Execution) are discovered from time to time by security researchers.

Mitigation with ASM

The Jackson-Databind developers are in a rush after new dangerous Java classes that may lead to remote code execution once deserialized and we keep on updating ASM with those Java deserialization gadgets.

As opposed to the Jackson-Databind developers we have additional mitigation layers to rely on, such as Java Server Side Code Injection signatures that detects operating system command execution attempts, for example by using Java's Runtime and ProcessBuilder classes (Signature ids 200003437,200003438,200003439, 200004174) and specific operating system command execution signatures.

Today, a new ASM security update was released to cover additional Java deserialization gadgets discovered recently:

- org.apache.tomcat.dbcp.dbcp2.BasicDataSource
- com.sun.org.apache.bcel.internal.util.ClassLoader
- org.hibernate.jmx.StatisticsService
- org.apache.ibatis.datasource.jndi.JndiDataSourceFactory
- org.springframework.context.support.FileSystemXmlApplicationContext

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com