

Spring Framework 远程代码执行漏洞 CVE-2022-22965



SANGFOR
深信服科技



深信服智安全
SANGFOR SECURITY

2022 年 3 月 31 日

一、漏洞概要

漏洞名称	Spring Framework 远程代码执行漏洞 CVE-2022-22965
发布时间	2022 年 3 月 31 日
组件名称	Spring Framework
影响范围	Spring Framework 5.3.X < 5.3.18 Spring Framework 5.2.X < 5.2.20 注：其他 Spring Framework 旧版本同样会受到影响。 Spring Framework 官方提供 Spring Framework 5.3.18 和 Spring Framework 5.2.20 两个安全版本（截止至 3 月 31 日）
漏洞类型	远程代码执行
利用条件	1、用户认证：不需要用户认证 2、前置条件：默认配置 3、触发方式：远程
综合评价	<综合评定利用难度>：容易，无需授权即可远程代码执行。 <综合评定威胁等级>：严重，能造成远程代码执行。

二、漏洞分析

2.1 组件介绍

Spring 是一个支持快速开发 Java EE 应用程序的框架。它提供了一系列底层容器和基础设施，并可以和大量常用的开源框架无缝集成，可以说是开发 Java EE 应用程序的必备。

2.2 漏洞描述

近日，深信服安全团队监测到一则 Spring Framework 组件存在远程代码执行漏洞的信息，漏洞威胁等级：严重。

该漏洞是由于 Spring Framework 未对传输的数据进行有效的验证，攻击者可利用该漏洞在未授权的情况下，构造恶意数据进行远程代码执行攻击，最终获取服务器最高权限。

三、影响范围

Spring Framework 作为主流的 Java 开发框架，全球有数百万使用 Spring Framework 框架的资产，可能受漏洞影响的资产广泛分布于世界各地，国内主要分布在广东、北京、上海等省市。

目前受影响的 Spring Framework 的版本：

Spring Framework 5.3.X < 5.3.18

Spring Framework 5.2.X < 5.2.20

注：其他 Spring Framework 旧版本同样会受到影响。

Spring Framework 官方提供

Spring Framework 5.3.18 和 Spring Framework 5.2.20 两个安全版本（截止至 3 月 31 日）

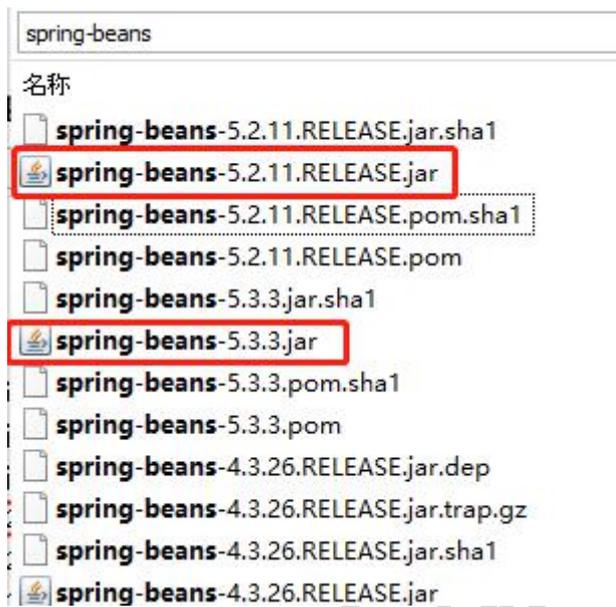
四、解决方案

4.1 修复建议

1. 如何检测组件版本

方案一

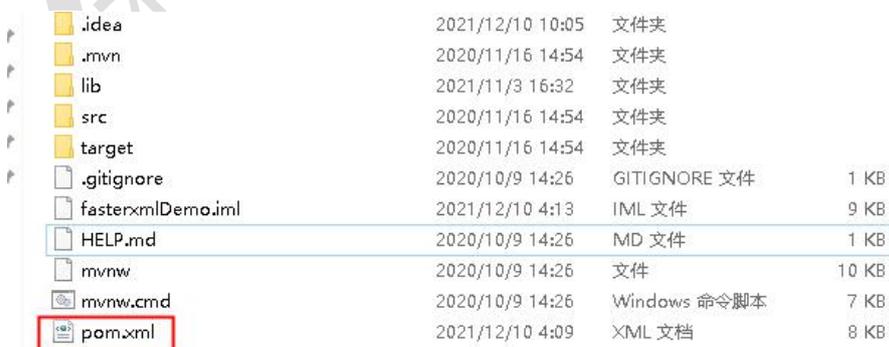
全盘搜索 spring-beans，如果存在 spring-beans-{version}.jar



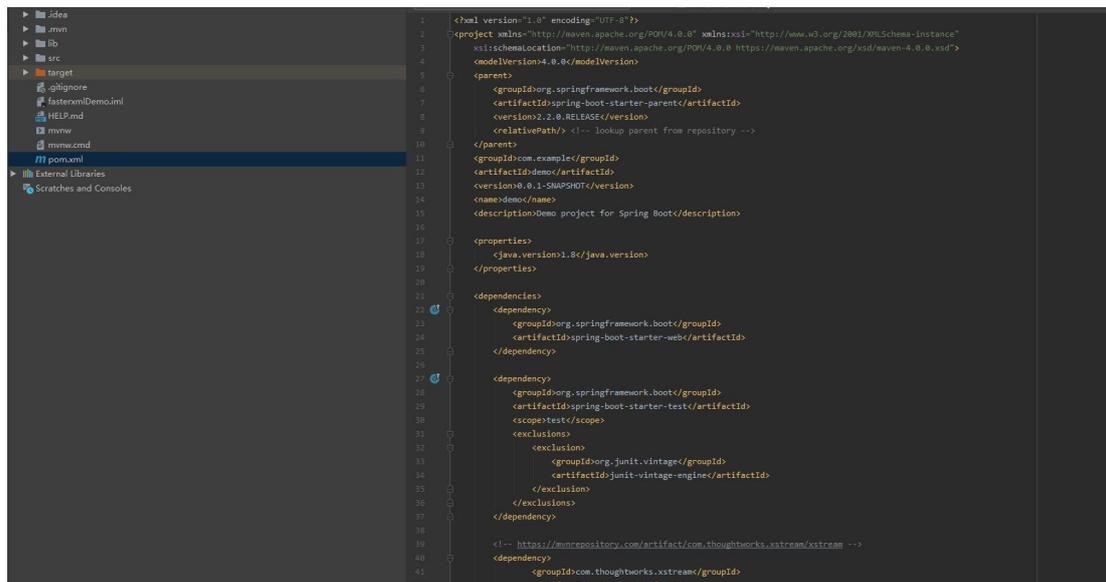
则用户可能受漏洞影响。

方案二

如果项目是由 maven 编译的（一般在项目根目录下会有 pom.xml）

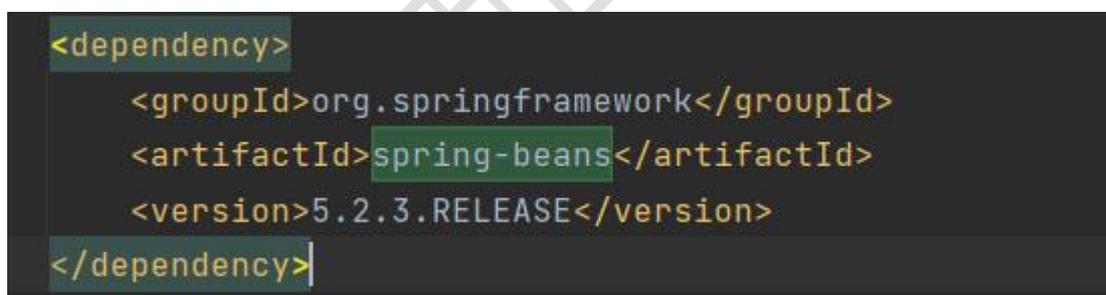


打开 pom.xml 文件，如图：



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4 <modelVersion>4.0.0</modelVersion>
5 <parent>
6 <groupId>org.springframework.boot</groupId>
7 <artifactId>spring-boot-starter-parent</artifactId>
8 <version>2.1.0.RELEASE</version>
9 <relativePath><!-- lookup parent from repository -->
10 </relativePath>
11 </parent>
12 <groupId>com.example</groupId>
13 <artifactId>demo</artifactId>
14 <version>0.0.1-SNAPSHOT</version>
15 <name>demo</name>
16 <description>Demo project for Spring Boot</description>
17
18 <properties>
19 <java.version>1.8</java.version>
20 </properties>
21
22 <dependencies>
23 <dependency>
24 <groupId>org.springframework.boot</groupId>
25 <artifactId>spring-boot-starter-web</artifactId>
26 </dependency>
27
28 <dependency>
29 <groupId>org.springframework.boot</groupId>
30 <artifactId>spring-boot-starter-test</artifactId>
31 <scope>test</scope>
32 <exclusions>
33 <groupId>org.junit.vintage</groupId>
34 <artifactId>junit-vintage-engine</artifactId>
35 </exclusions>
36 </dependency>
37
38 <!-- https://mvnrepository.com/artifact/com.thoughtworks.xstream/xstream -->
39 <dependency>
40 <groupId>com.thoughtworks.xstream</groupId>
41 <artifactId>xstream</artifactId>
42 </dependency>
43 </dependencies>
44 </project>
```

在此文件中搜索 spring-beans，如果可以搜索到关键字，并且<version>标签内部的字段小于 5.3.18 或者 5.2.20，则可能受到漏洞的影响。（图中的 spring-beans 的版本是 5.2.3，在漏洞影响范围内）



```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
  <version>5.2.3.RELEASE</version>
</dependency>
```

如以上检索均未发现结果，不能够完全下结论一定没有使用 Spring Framework 框架。

2.官方修复建议

当前 Spring Framework 官方已发布最新版本, 建议受影响的用户及时更新升级到最新版本。链接如下:

<https://github.com/spring-projects/spring-framework/tags>

注: Spring Framework 5.3.18 和 Spring Framework 5.2.20 是 Spring 官方提供的两个安全版本 (截止至 3 月 31 日)

3.临时修复建议

该临时修复建议存在一定风险, 建议用户可根据业务系统特性谨慎选择采用临时修复方案:

需同时按以下两个步骤进行漏洞的临时修复:

1. 在应用中全局搜索@InitBinder 注解, 看看方法体内是否调用 dataBinder.setDisallowedFields 方法, 如果发现此代码片段的引入, 则在原来的黑名单中, 添加{"class.*", "Class.*", "*.class.*", "*.Class.*"}。(注:如果此代码片段使用较多,需要每个地方都追加)

2. 在应用系统的项目包下新建以下全局类, 并保证这个类被 Spring 加载到(推荐在 Controller 所在的包中添加).完成类添加后, 需对项目进行重新编译打包和功能验证测试。并重新发布项目。

```
import org.springframework.core.annotation.Order;
```

```
import org.springframework.web.bind.WebDataBinder;
```

```
import org.springframework.web.bind.annotation.ControllerAdvice;

import org.springframework.web.bind.annotation.InitBinder;

@ControllerAdvice

@Order(10000)

public class GlobalControllerAdvice{

    @InitBinder

    public void setAllowedFields(webdataBinder dataBinder){

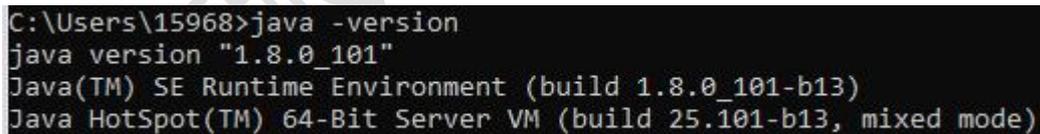
        String[]abd=new string[]{"class.*","Class.*","*.class.*","*.Class.*"};

        dataBinder.setDisallowedFields(abd);

    }

}
```

注：在业务系统的运行服务器上，执行“java -version”命令查看运行的 JDK 版本，如果版本号小于等于 8，则不受漏洞影响，如图所示：



```
C:\Users\15968>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
```

上图查找的 JDK 版本为 8，不受漏洞影响。

4.2 深信服解决方案

1.主动检测

支持对 **Spring Framework** 远程代码执行漏洞的主动检测，可批量快速检出业务场景中该事件的受影响资产情况，相关产品如下：

【深信服安全云眼 **CloudEye**】已发布解决方案。

【深信服云镜 **YJ**】已发布解决方案。

【深信服主机安全保护平台 **CWPP**】已发布解决方案。

【深信服安全托管服务 **MSS**】已发布解决方案，提供漏洞自查脚本，下载链接如下：

<https://bbs.sangfor.com.cn/forum.php?mod=viewthread&tid=204347&page=1&extra=#/pid2565591>

2. 安全监测

支持对 **Spring Framework** 远程代码执行漏洞的监测，可收集实时监控业务场景中的受影响资产情况，快速检查受影响范围，相关产品及服务如下：

【深信服安全感知管理平台 **SIP**】已发布解决方案

【深信服终端安全管理系统 **EDR**】已发布解决方案

【深信服安全托管服务 **MSS**】已发布解决方案

3. 安全防护

支持对 **Spring Framework** 远程代码执行漏洞的防御，可阻断攻击

者针对该事件的入侵行为，相关产品及服务如下：

【深信服下一代防火墙 AF】已发布解决方案。

【深信服 Web 应用防火墙 WAF】已发布解决方案。

【深信服安全托管服务 MSS】已发布解决方案。

五、时间轴

2022/3/29 深信服监测到 Spring Framework 远程代码执行漏洞信息。

2022/3/31 深信服千里目安全实验室发布漏洞通告,并发布解决方案。

六、参考链接

<https://spring.io/blog/2022/03/31/spring-framework-rce-early-announcement>

七、了解更多

深信服千里目安全实验室持续紧跟国内外漏洞威胁情报，从中筛选出能给客户带来威胁的漏洞，第一时间推送解决方案，持续提供可感知的安全感。在这场永不停歇的攻防战争中，深信服千里目安全实验室掌握一手漏洞情报，坚持“千里之外，洞悉风险”，与各大网络

安全厂商一同维护网络安全，构建平衡、和谐的网络生态系统。关注深信服千里目安全实验室微信公众号，第一时间了解更多漏洞情报。



深信服千里目安全实验室