



Spring4Shell

ZEROFOX INTELLIGENCE

F-2022-04-01a

Classification: TLP White

Criticality: High

Intelligence Requirements: Vulnerability

April 1, 2022

Scope Note

*ZeroFox Intelligence is derived from a variety of sources, including—but not limited to—curated open-source accesses, vetted social media, proprietary data sources, and direct access to threat actors and groups through covert communication channels. Information relied upon to complete any report cannot always be independently verified. As such, ZeroFox applies rigorous analytic standards and tradecraft in accordance with best practices and includes caveat language and source citations to clearly identify the veracity of our Intelligence reporting and substantiate our assessments and recommendations. All sources used in this particular Intelligence product were **identified prior to 9:00 AM (EDT) on April 1, 2022**; per cyber hygiene best practices, caution is advised when clicking on any third-party links.*

Spring4Shell

Key Findings

- A zero-day vulnerability dubbed “Spring4Shell” enables remote code execution in the Spring Core Framework.
- The vulnerability impacts Spring Model-View-Controller (MVC) and Spring WebFlux applications running with certain configurations.
- The vulnerability, now tracked as CVE-2022-22965, is caused by unsafe deserialization of passed arguments.
- Patches have been released to address vulnerabilities in versions of Spring Framework (5.3.18 and 5.2.20) and Spring Boot (2.5.12 and 2.6.6).

Analyst Commentary

“Spring4Shell” or “SpringShell” is a zero-day remote code execution (RCE) vulnerability in the Spring Core Framework, a widely-used application framework that enables software developers to create Java applications; reporting indicates that approximately 70 percent of all Java applications rely on it. Now tracked as CVE-2022-22965 and given a CVSS score of 9.8 out of 10, this zero-day was first disclosed on March 30, 2022, by an alleged researcher. A successful exploitation leveraging this vulnerability allows threat actors to execute arbitrary code. Given such widespread usage of this framework, threat actors can take advantage of automation to rapidly check for and exploit this vulnerability against a multitude of targets.

Analysis indicates this vulnerability is caused by unsafe deserialization of passed arguments to the `getCacheIntrospectionResults` method. This method erroneously exposes a class object when binding these parameters. Under certain configurations, the vulnerability can be exploited by sending a specially-crafted HTTP request. Exploit code has been made publicly available. This vulnerability impacts Spring Model-View-Controller (MVC) and Spring WebFlux applications running with certain configurations. Existing exploitation proofs of concept have worked under the following conditions:

- JDK 9 or higher
- Apache Tomcat as the Servlet container
- Packaged as a traditional WAR (in contrast to a Spring Boot executable jar)
- spring-webmvc or spring-webflux dependency
- Spring Framework versions 5.3.0 to 5.3.17, 5.2.0 to 5.2.19 and older versions

Any Java application using the Spring Beans packet and using Spring parameters binding could be affected by this vulnerability. Reporting also indicates that exploitation requires an endpoint with DataBinder enabled. Reporting further indicates that sample code from spring[.]io is also vulnerable, which may have been used by developers as a template for their applications.

On March 31, 2022, Spring released emergency patches 5.3.18 and 5.2.20 to fix this vulnerability within versions 5.3 and 5.2. Patches have also been released for Spring Boot (which depends on the Spring Framework) 2.5.12 and 2.6.6.

ZeroFox continues to anticipate increased targeting of vulnerabilities in commonly-used software modules and off-the-shelf packages. The name “Spring4Shell” is designed to highlight this vulnerability’s similarities to the Log4j (CVE-2021-42278) vulnerability that was disclosed in December 2021. Threat actors were able to execute a specially-crafted string against the vulnerable and widely-used Java logging library, thereby compromising numerous hosts. While the patch to the Log4j library was quickly released to mitigate against the vulnerability, the Log4j exploit remains a critical threat months later due to the time taken to update vulnerable devices.

The “Spring4Shell” vulnerability follows a separate RCE vulnerability (CVE-2022-22963) impacting Spring Cloud (SPEL) with a CVSS score of 9.0 out of 10. On March 29, 2022, this vulnerability was patched with the release of Spring Cloud Function 3.1.7 and 3.2.3.

Recommendations

- Identify all external-facing devices and applications that utilize JDK9+ and the Spring Framework.
- Patch Spring Core Framework to v.5.3.18 and v.5.2.20 where possible.
- Patch Spring Boot to v.2.5.12 and 2.6.6 where possible.

Sources

- [hXXps://spring\[.\]io/blog/2022/03/31/spring-framework-rce-early-announcement](https://spring.io/blog/2022/03/31/spring-framework-rce-early-announcement)
- [hXXps://securityboulevard\[.\]com/2022/03/spring4shell-what-happened-whos-vulnerable-and-how-to-mitigate/](https://securityboulevard[.]com/2022/03/spring4shell-what-happened-whos-vulnerable-and-how-to-mitigate/)
- [hXXps://www.praetorian\[.\]com/blog/spring-core-jdk9-rce/](https://www.praetorian[.]com/blog/spring-core-jdk9-rce/)
- [hXXps://ot.alienvault\[.\]com/pulse/6246c5778ca27726b90d842e](https://ot.alienvault[.]com/pulse/6246c5778ca27726b90d842e)

Indicators of Compromise

IPs

- 38.83.79[.]203
- 103.214.146[.]5
- 149.28.147[.]15
- 158.247.202[.]6

Filename

- wpz.jsp



Appendix: Traffic-Light Protocol for Information Dissemination

Red

WHEN SHOULD IT BE USED?

Sources may use TLP:

RED when information cannot be effectively acted upon by additional parties and could lead to impacts on a party's privacy, reputation, or operations if misused.

HOW MAY IT BE SHARED?

Recipients may NOT share TLP:

RED with any parties outside of the specific exchange, meeting, or conversation in which it is originally disclosed.

Amber

Sources may use TLP:

AMBER when information requires support to be effectively acted upon but carries risks to privacy, reputation, or operations if shared outside of the organizations involved.

Recipients may ONLY share TLP:

AMBER information with members of their own organization and only as widely as necessary to act on that information.

Green

WHEN SHOULD IT BE USED?

Sources may use TLP:

GREEN when information is useful for the awareness of all participating organizations, as well as with peers within the broader community or sector.

HOW MAY IT BE SHARED?

Recipients may share TLP:

GREEN information with peers & partner organizations within their sector or community but not via publicly accessible channels.

White

Sources may use TLP:

WHITE when information carries minimal or no risk of misuse in accordance with applicable rules and procedures for public release.

TLP:

WHITE information may be distributed without restriction, subject to copyright controls.