Incidents

Spring4Shell (CVE-2022-22965): details and mitigations

Incidents

04 Apr 2022

1 minute read

Table of Contents

Authors

- AMR

Last week researchers found the critical vulnerability CVE-2022-22965 in Spring — the open source Java framework. Using the vulnerability, an attacker can execute arbitrary code on a remote web server, which makes CVE-2022-22965 a critical threat, given the Spring framework's popularity. By analogy with the infamous Log4Shell threat, the vulnerability was named Spring4Shell.

CVE-2022-22965 and CVE-2022-22963: technical details

CVE-2022-22965 (Spring4Shell, SpringShell) is a vulnerability in the Spring Framework that uses data binding functionality to bind data stored within an HTTP request to certain objects used by an application. The bug exists in the getCachedIntrospectionResults method, which can be used to gain unauthorized access to such objects by passing their class names via an HTTP request. It creates the risks of data leakage and remote code execution when special object classes are used. This vulnerability is similar to the long-closed CVE-2010-1622, where class name checks were added as a fix so that the name did not match classLoader or protectionDomain. However, in a newer version of JDK an alternative method exists for such exploitation, for example, through Java 9 Platform Module System functionality. So an attacker can overwrite the Tomcat logging configuration and then upload a JSP web shell to execute arbitrary commands on a server running a vulnerable version of the framework.

A vulnerable configuration consists of:

- JDK version 9+
- Apache Tomcat for serving the application
- Spring Framework versions 5.3.0 to 5.3.17 and 5.2.0 to 5.2.19 and below
- application built as a WAR file

CVE-2022-22963 is a vulnerability in the routing functionality of Spring Cloud Function that allows code injection through Spring Expression Language (SpEL) by adding a special spring.cloud.function.routing-expression header to an HTTP request. SpEL is a special expression language created for Spring Framework that supports queries and object graph management at runtime. This vulnerability can also be used for remote code execution.

A vulnerable configuration consists of:

- Spring Cloud Function 3.1.6, 3.2.2 and older versions

Mitigations for Spring vulnerabilities exploitation

CVE-2022-22965 is fixed in 2.6.6; see the Spring blog for details.

To fix CVE-2022-22963, you also need to install the new Spring Cloud Function versions; see the VMware website for details.

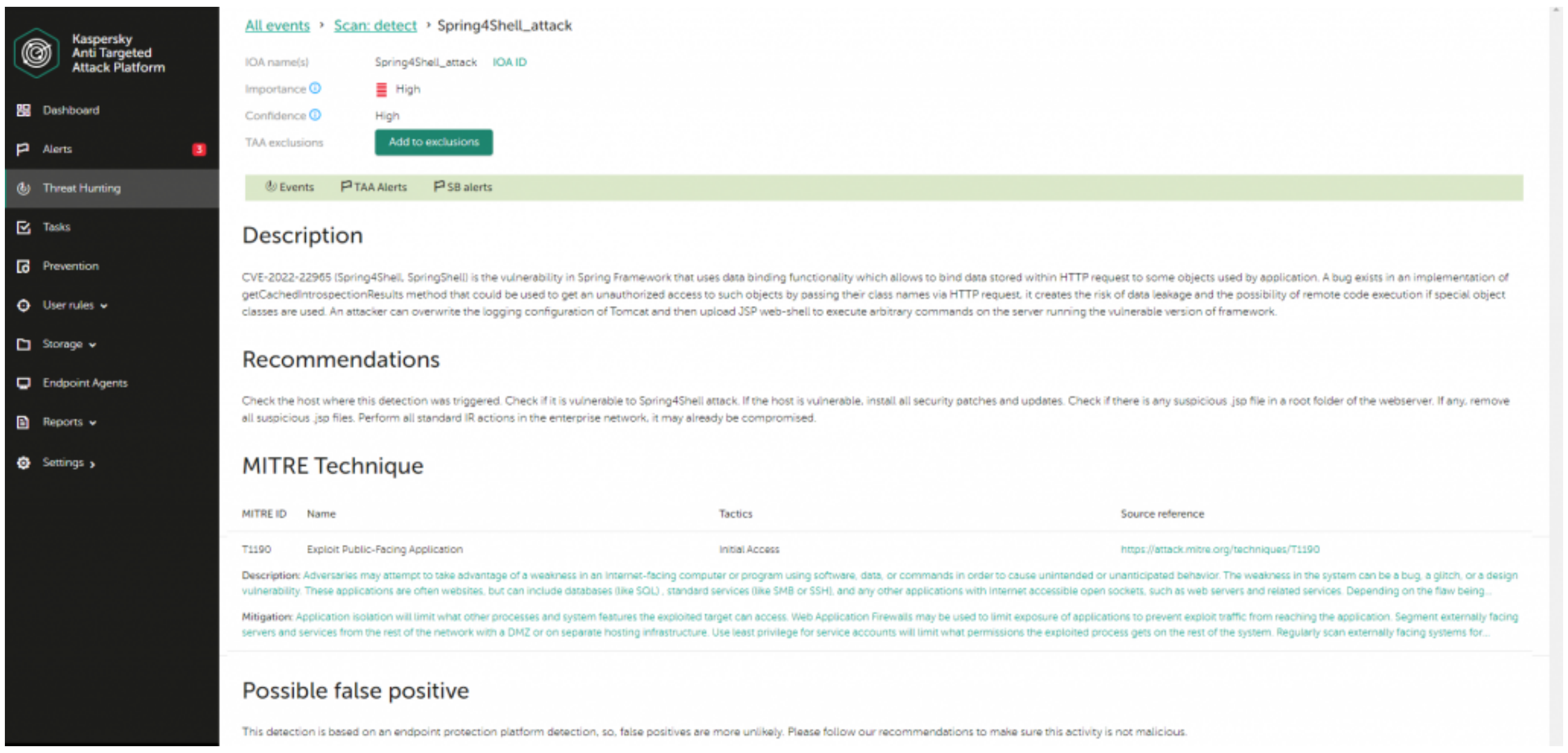
To detect exploitation attempts, ensure that Advanced Exploit Prevention and Network Attack Blocker features are enabled. Some techniques used during exploitation can be seen in other exploits that we detect, which is why the verdict names can differ.

Indicators of Compromise

Verdicts PDM:Exploit.Win32.Generic UMIDS:Intrusion.Generic.Agent.gen Intrusion.Generic.CVE-*.*

MD5 hashes of the exploits 7e46801dd171bb5bf1771df1239d760c — shell.jsp (CVE-2022-22965) 3de4e174c2c8612aebb3adef10027679 — exploit.py (CVE-2022-22965)

Detection of the exploitation process with Kaspersky EDR Expert



IOA name(s)	Spring4Shell_attack IOA ID
Importance	High
Confidence	High
TAA exclusions	Add to exclusions

Description

CVE-2022-22965 (Spring4Shell, SpringShell) is the vulnerability in Spring Framework that uses data binding functionality which allows to bind data stored within HTTP request to some objects used by application. A bug exists in an implementation of getCachedIntrospectionResults method that could be used to get an unauthorized access to such objects by passing their class names via HTTP request. it creates the risk of data leakage and the possibility of remote code execution if special object classes are used. An attacker can overwrite the logging configuration of Tomcat and then upload JSP web-shell to execute arbitrary commands on the server running the vulnerable version of framework.

Recommendations

Check the host where this detection was triggered. Check if it is vulnerable to Spring4Shell attack. If the host is vulnerable, install all security patches and updates. Check if there is any suspicious .jsp file in a root folder of the webserver. If any, remove all suspicious .jsp files. Perform all standard IR actions in the enterprise network, it may already be compromised.

MITRE Technique

MITRE ID	Name	Tactics	Source reference
T1190	Exploit Public-Facing Application	Initial Access	https://attack.mitre.org/techniques/T1190

Description: Adversaries may attempt to take advantage of a weakness in an Internet-facing computer or program using software, data, or commands in order to cause unintended or unanticipated behavior. The weakness in the system can be a bug, a glitch, or a design vulnerability. These applications are often websites, but can include databases (like SQL), standard services (like SMB or SSH), and any other applications with Internet accessible open sockets, such as web servers and related services. Depending on the flaw being...

Mitigation: Application isolation will limit what other processes and system features the exploited target can access. Web Application Firewalls may be used to limit exposure of applications to prevent exploit traffic from reaching the application. Segment externally facing servers and services from the rest of the network with a DMZ or on separate hosting infrastructure. Use least privilege for service accounts will limit what permissions the exploited process gets on the rest of the system. Regularly scan externally facing systems for...

Possible false positive

This detection is based on an endpoint protection platform detection, so, false positives are more unlikely. Please follow our recommendations to make sure this activity is not malicious.

- Java
- Malware Descriptions
- Vulnerabilities and exploits

Authors

- AMR