1- Summary

We can say that Spring, which is one of the framework structures mostly used in Java applications, is basically a framework that organizes the development process. Spring4Shell is defined as a vulnerability that comes from making good use of the ability to bind various parameters to objects in applications developed using the Spring Framework. Spring4Shell allows the attacker to execute remote commands, RCE (Remote Code Execution), under certain conditions, on the inputs sent by the attacker. The vulnerability is tracked as CVE-2022-22965 and is also known as “Spring4Shell” or “SpringShell”.

2- Introduction

CVSS score of the vulnerability is a 9.8 Critical, CVSS vector is CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H. A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

3- Impact

This vulnerability cause of Unauthorized Access and Information Disclosure. Attacker might access the critic data even she/he is unauthorized.

4- Exploit explanation

Threat actors can directly access an object by specifying the class variable in their requests. All child properties of an object can also be accessed by malicious actors through the class objects. As a result, they can get access to all kinds of other valuable objects on the system simply by following the chains of properties.

Having access to the class variable and all its sub-properties provides a path for threat actors to change the behavior of the web application. Their familiarity with ways to exploit exposed class objects has resulted in many techniques for weaponizing this vulnerability.

For example, threat actors can access an AccessLogValve object and weaponize the class variable "class.module.classLoader.resources.context.parent.pipeline.firstpath" in Apache Tomcat. They can do this by redirecting the access log to write a web shell into the web root through manipulation of the properties of the AccessLogValve object, such as its pattern, suffix, directory, and prefix.

5- Current Explotation Status

We observed active exploitation of Spring4Shell wherein malicious actors were able to weaponize and execute the Mirai botnet malware on vulnerable servers, specifically in the Singapore region.

İf you want to watch PoC follow this link:

https://github.com/me2nuk/CVE-2022-22965

6- Mitigation Suggestions

- The vulnerability can be avoided by upgrading the Spring Framework to version 5.3.18 and 5.2.20 or upgrading to a higher version.
- For older applications, upgrading to Apache Tomcat 10.0.20, 9.0.62 or 8.5.78, running on an unsupported version of Spring Framework on Tomcat, provides adequate protection.
- If you cannot upgrade Spring Framework or upgrade Apache Tomcat, switching to Java 8 is a viable workaround

7- Conclusion

We are unable confirm if the exploitation attempts we analyzed for this blog entry were successful. It should be noted that we also observed Linux payloads where the script ldr.sh attempts to stop other running cryptocurrency miners to run its own payload.

8- Referances

- https://www.acunetix.com/blog/web-security-zone/critical-alert-spring4shell-rce-cve-2022-22965-in-spring
- https://www.rapid7.com/blog/post/2022/03/30/spring4shell-zero-day-vulnerability-in-spring-framework
- https://spring.io/blog/2022/03/31/spring-framework-rce-early-announcement#vulnerability
- https://securelist.com/spring4shell-cve-2022-22965/106239
- https://cert-portal.siemens.com/productcert/pdf/ssa-254054.pdf
- https://psirt.global.sonicwall.com/vuln-detail/SNWLID-2022-0005
- https://www.trendmicro.com/en_us/research/22/d/cve-2022-22965-analyzing-the-exploitation-of-spring4shell-vulner.html#:~:text=We%20discovered%20active%20exploitation%20of,download%20the%20Mirai%20botnet%20malware.