

SWRhapsody

CVE-2020-5410

于6月 3, 2020由SWRhapsody发布

Introduction

补天挖的 spring-cloud-config 目录穿越^[1]

Code

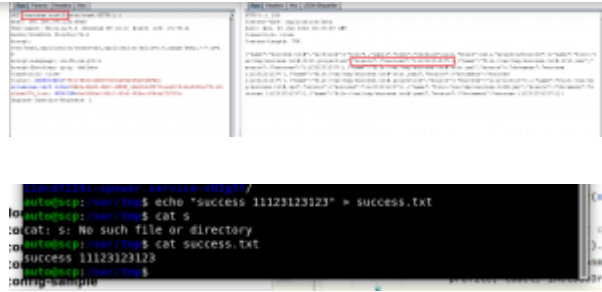
安装

```
1 https://github.com/spring-cloud/spring-cloud-config
2 git checkout bbb5f19ce219e9dfef01a45f3e576f161dfb2685
```

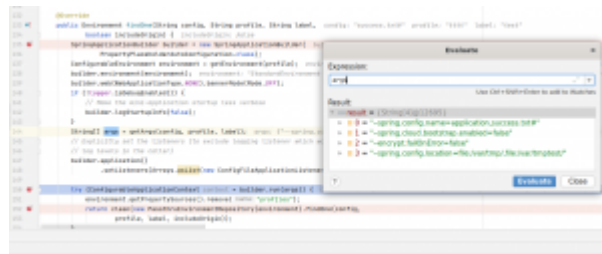
修改 spring-cloud-config-server/src/main/resources/configserver.yml 这个文件

```
1 info:
2   component: Config Server
3 spring:
4   profiles:
5     active: native
6   application:
7     name: configserver
8   autoconfigure.exclude: org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfigurati
9   jmx:
10    default_domain: cloud.config.server
11   cloud:
12     config:
13       server:
14         native:
15           # search-locations: file://${HOME}/Desktop/config
16           search-locations: file:/var/tmp
17         git:
18           uri: https://github.com/spring-cloud-samples/config-repo
19           repos:
20             - patterns: multi-repo-demo-*
21               uri: https://github.com/spring-cloud-samples/config-repo
22
23   server:
24     port: 8888
25   management:
26     context_path: /admin
```

SWRhapsody

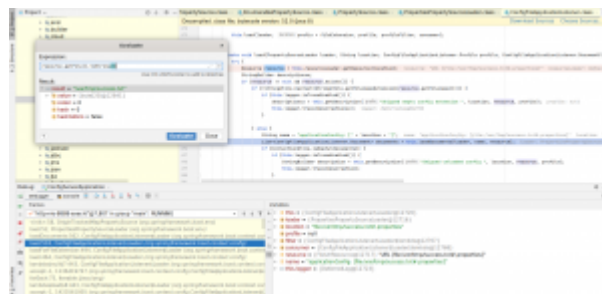


这个漏洞比较有意思，首先发送poc后会进入到 `spring-cloud-config-server/src/main/java/org/springframework/cloud/config/server/environment/NativeEnvironmentRepository.java#findOne` 中



Spring-cloud-config 会将 “`success.txt#`” 通过 “`-spring.config.name`” 传给 spring boot，让其加载 “`success.txt#.properties`” 等文件的内容到环境变量中。

然后将这些放在环境变量中的配置返回给用户，但是由于 `#` 的存在，spring boot 会加载拼接后的配置文件也就是 `success.txt#.properties`，接着 spring boot 在 `.m2/repository/org/springframework/boot/spring-boot/2.2.5.RELEASE/spring-boot-2.2.5.RELEASE.jar!/org/springframework/boot/context/config/ConfigFileApplicationListener.class` 文件中加载这个文件



可以看到 `location` 虽然是 `/var/tmp/success.txt#.properties` 但加载出的文件却是 `/var/tmp/success.txt`，也就是说 `resourceLoader.getResource` 加载文件路径的时候忽略了 `#` 号后面的内容。

如果是正常的请求就是此处的 `location` 会是 `/var/tmp/success.txt.properties`，这种情况下会找不

SWRhapsody

Slient Fix

下面这是个 slient fix 的漏洞

安装

```
1 https://github.com/spring-cloud/spring-cloud-config
2 git checkout c9a9f77a2b6d57272f565b5472ab88263a1df1d7
```

修改 spring-cloud-config-server/src/main/resources/configserver.yml 这个文件

```
1 info:
2   component: Config Server
3 spring:
4   profiles:
5     active: native
6   application:
7     name: configserver
8   autoconfigure.exclude: org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration
9   jmx:
10    default_domain: cloud.config.server
11   cloud:
12     config:
13       server:
14         native:
15           # search-locations: file://${HOME}/Desktop/config
16           search-locations: file:./
17         git:
18           uri: https://github.com/spring-cloud-samples/config-repo
19           repos:
20             - patterns: multi-repo-demo-*
21               uri: https://github.com/spring-cloud-samples/config-repo
22
23   server:
24     port: 8888
25   management:
26     context_path: /admin
```

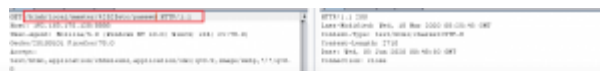
注意

```
1 search-locations: file:./
```

接下来直接看补丁的 test [2]



所以 POC 应该是



SWRhapsody

名称 *

电子邮件 *

网站

在想些什么?

发表评论

近期文章

[携程Apollo YAML 反序列化](#)

[CVE-2020-5410](#)

[CodeQL部分源码简读](#)

[服务器与网关的不一致](#)

[CodeQL 部分使用记录](#)

近期评论

文章归档

2020年8月

SWRhapsody

2020年5月

2020年3月

2020年1月

2019年12月

2019年11月

2019年8月

2019年7月

2019年5月

2019年4月

2019年1月

2018年11月

2018年10月

2018年9月

2018年4月

2018年3月

2018年2月

2018年1月

分类目录

Article Collection

Cheat Sheet

cryptography

Exercise

Exploit

SWRhapsody

Uncategorized

相关文章

EXPLOIT

携程Apollo YAML 反序列化

Introduction 3月份发现的一个问题，7月份提交给的携程SR [阅读更多...](#)

EXPLOIT

CVE-2020-1957

Introduction 这个漏洞需要1.5.2 版本以下的 shir [阅读更多...](#)

EXPLOIT

CVE-2020-5405

Introduction 记得我刚开始写博客的时候是无比兴奋的，觉得终 [阅读更多...](#)

SWRhapsody
