Spring Cloud Config Server 路径穿越与任意文件读取漏洞分析 - 【CVE-2019-3799】

chybeta / 2019-04-18 08:02:00 / 浏览数 8094 安全技术 漏洞分析 顶(0) 踩(0)

漏洞公告

https://pivotal.io/security/cve-2019-3799

CVE-2019-3799: Directory Traversal with spring-cloud-config-server

Severity

High

Vendor

Spring by Pivotal

Description

Spring Cloud Config, versions 2.1.x prior to 2.1.2, versions 2.0.x prior to 2.0.4, and versions 1.4.x prior to 1.4.6, and older unsupported versions allow applications to serve arbitrary configuration files through the spring-cloud-config-server module. A malicious user, or attacker, can send a request using a specially crafted URL that can lead a directory traversal attack.

Affected Pivotal Products and Versions

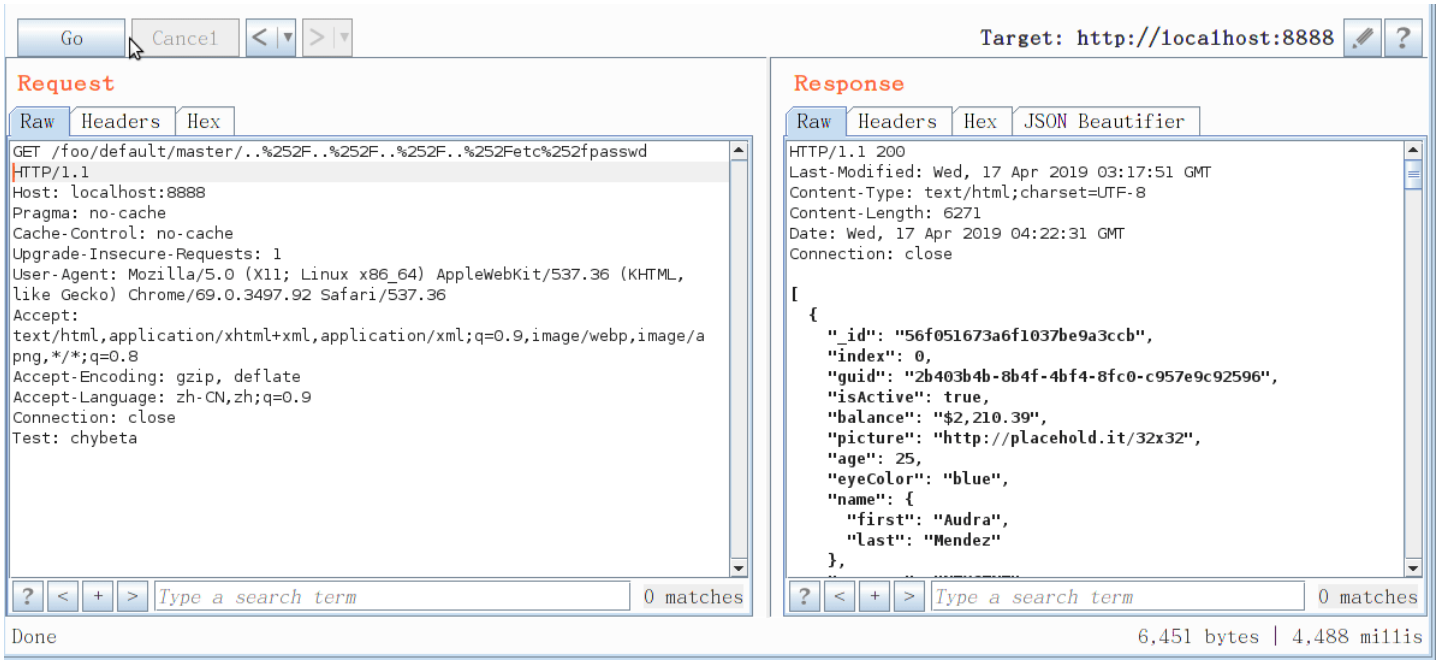
Severity is high unless otherwise noted.

- Spring Cloud Config 2.1.0 to 2.1.1
- Spring Cloud Config 2.0.0 to 2.0.3
- Spring Cloud Config 1.4.0 to 1.4.5
- Older unsupported versions are also affected

漏洞复现

环境搭建： https://github.com/spring-cloud/spring-cloud-config#quick-start

```
GET /foo/default/master/..%252F..%252F..%252F..%252Fetc%252fpasswd HTTP/1.1
Host: localhost:8888
```

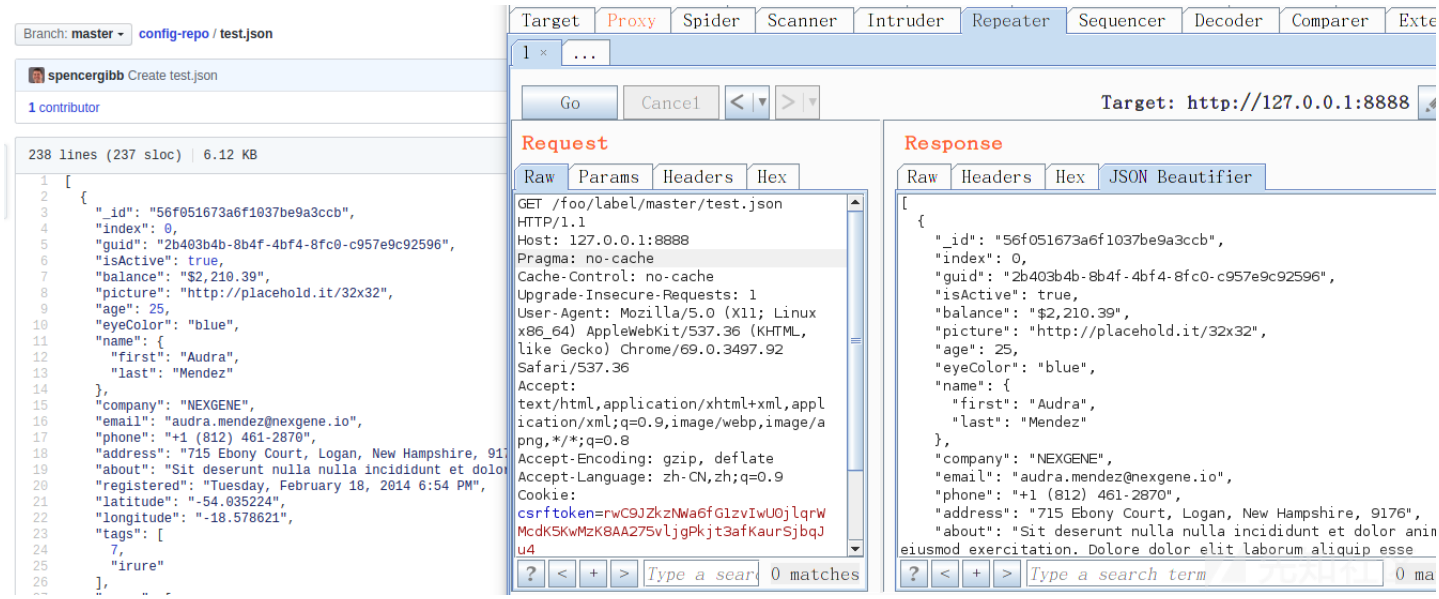


漏洞分析

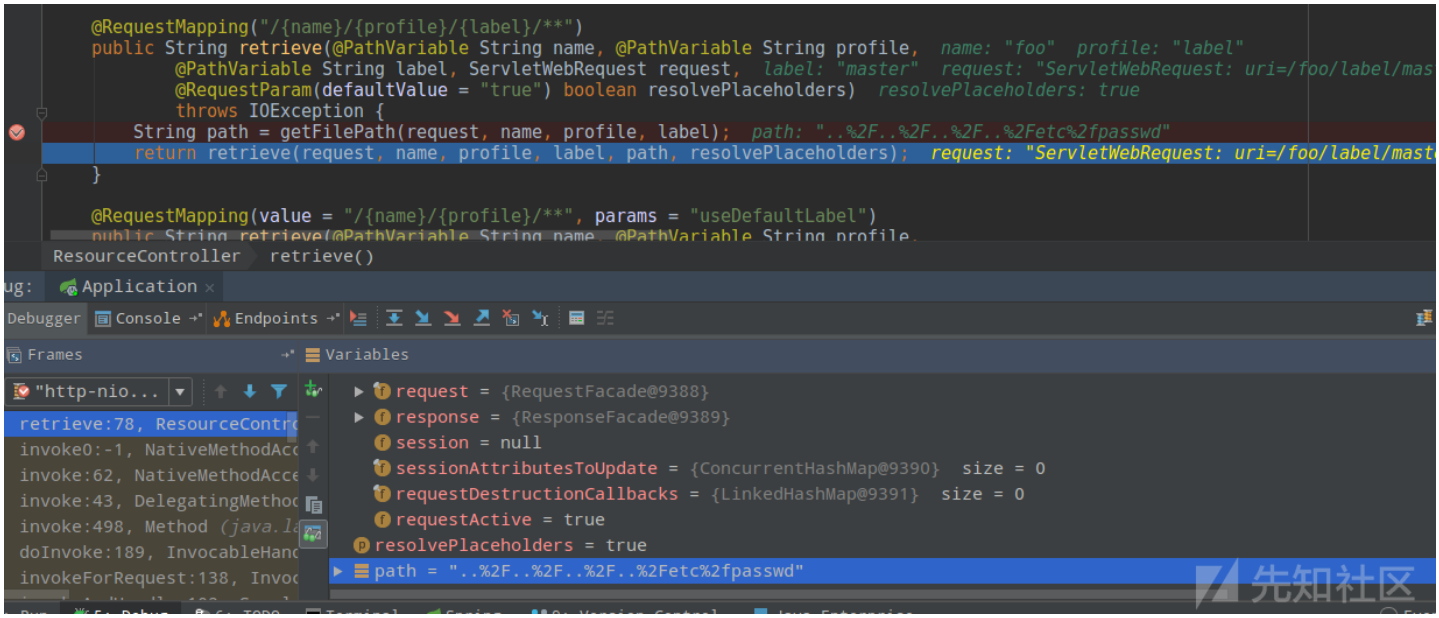
Spring Cloud Config是Spirng Cloud下用于分布式配置管理的组件，分为Config-Server和Config-Client两个角色。
Config-Server负责集中存储/管理配置文件，Config-Client则可以从Config-Server提供的HTTP接口获取配置文件使用。2019年4月16日，Pivotal官方发布安全通
Cloud Config Server 部分版本存在目录遍历漏洞，据此可以获取Server端服务器文件。

根据官方文档，可以通过如下请求GET /{name}/{profile}/{label}/{path}
来获取配置文件，name，profile和label的含义与常规环境下的endpoint相同，而path是指文件名。以官方示例为环境，我们请求
https://github.com/spring-cloud-samples/config-repo/blob/master/test.json 这个文件并以文本形式返回 ，则我们需要向Spring Cloud Config
Server发出如下请求：

```
GET http://127.0.0.1:8888/foo/label/master/test.json
```



根据请求格式可以在 org/springframework/cloud/config/server/resource/ResourceController.java:54 中找到对应的处理
@RequestMapping("/{name}/{profile}/{label}/**")：

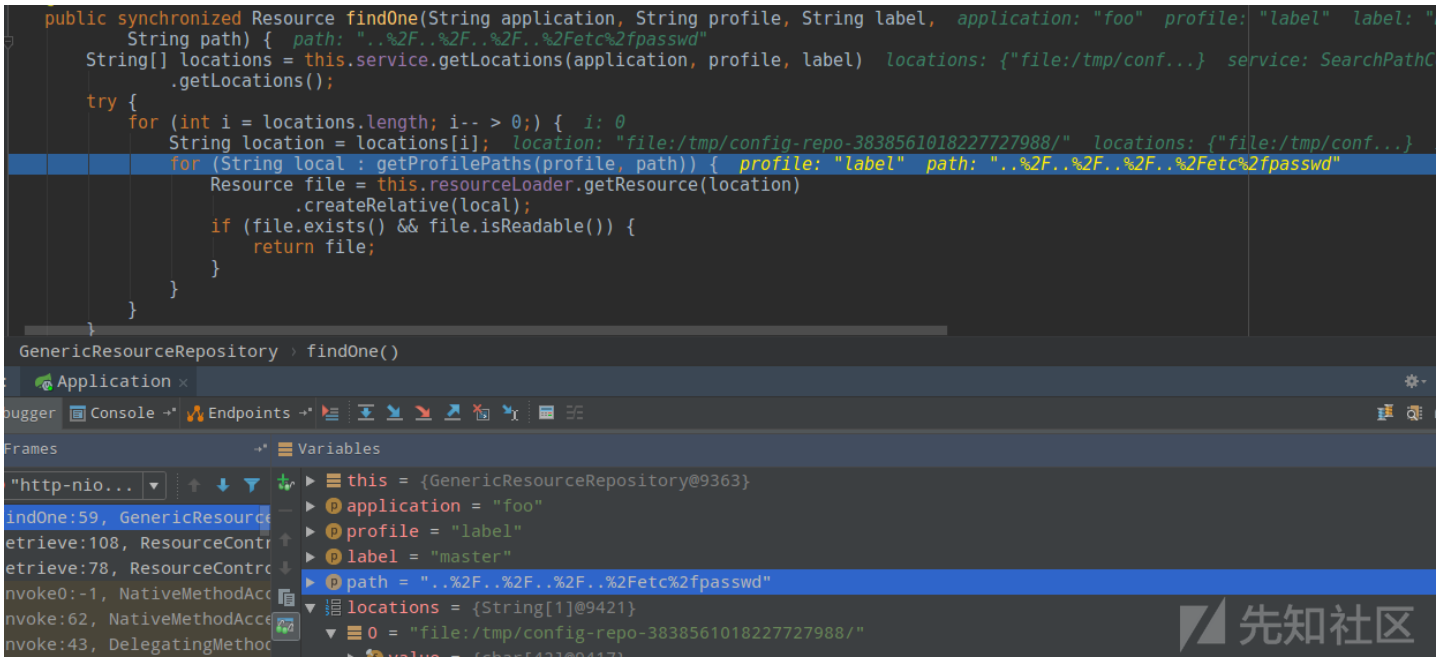


其中path值即为payload:`..%2F..%2F..%2F..%2Fetc%2fpasswd`

跟入`retrieve` 在`org/springframework/cloud/config/server/resource/ResourceController.java:104` :

```
synchronized String retrieve(ServletWebRequest request, String name, String profile,
            String label, String path, boolean resolvePlaceholders) throws IOException {
        name = resolveName(name);
        label = resolveLabel(label);
        Resource resource = this.resourceRepository.findOne(name, profile, label, path);
        ...
    }
```

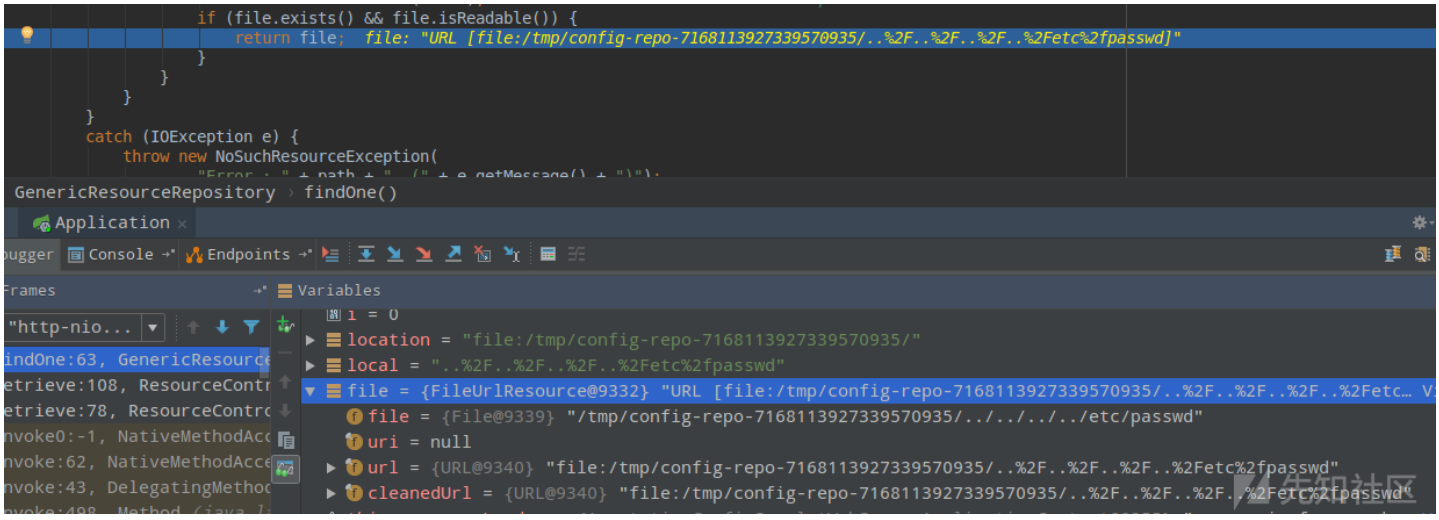
这里会根据前面所传条件获取到resource。文档中提到`only the first one to match is returned`，所以继续跟入`findOne`:



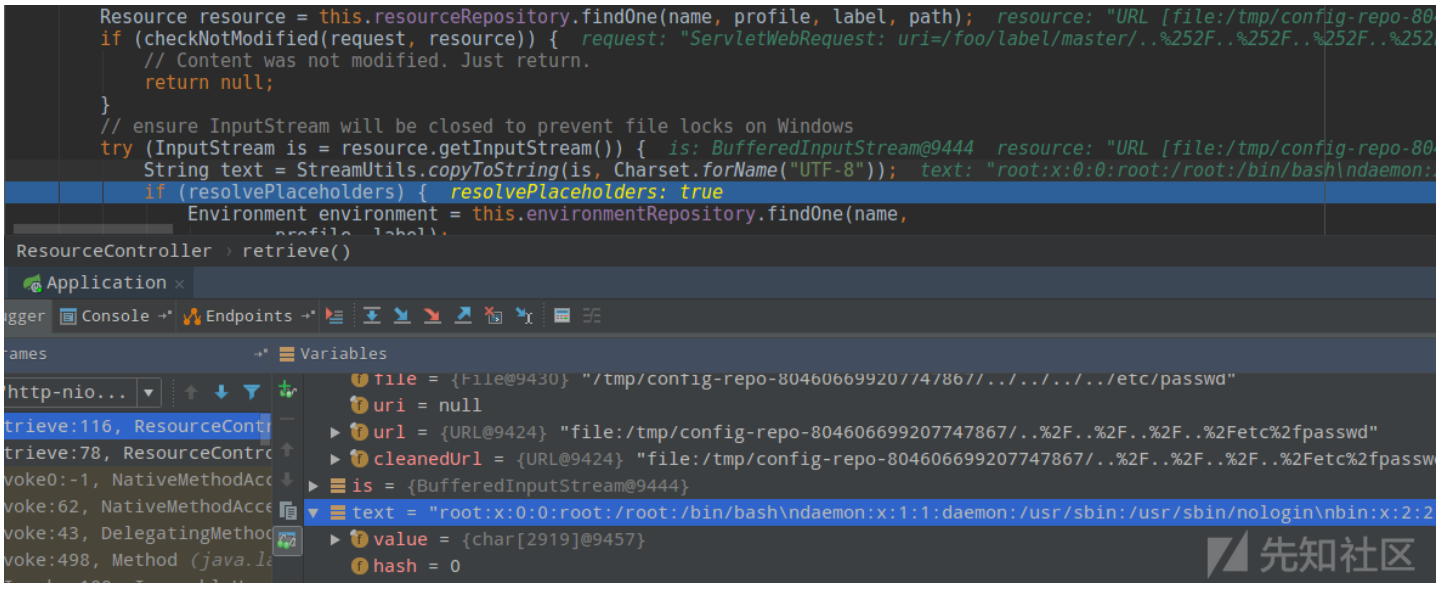
可以看到这里`locations`的值为`file:/tmp/config-repo-7168113927339570935/`，这是`Config-Server`从后端拉取到配置文件时临时存放，正常情况下将会在该

```
➜  config-repo-7168113927339570935 git:(master) pwd
/tmp/config-repo-7168113927339570935
➜  config-repo-7168113927339570935 git:(master) git remote -v
origin  https://github.com/spring-cloud-samples/config-repo (fetch)
origin  https://github.com/spring-cloud-samples/config-repo (push)
➜  config-repo-7168113927339570935 git:(master) cat test.json
[
  {
    "_id": "56f051673a6f1037be9a3ccb",
    "index": 0,
    "guid": "2b403b4b-8b4f-4bf4-8fc0-c957e9c92596",
    "isActive": true,
    "balance": "$2,210.39",
    "picture": "http://placehold.it/32x32",
    "age": 25,
    "eyeColor": "blue",
    "name": {
      "first": "Audra",
      "last": "Mendez"
    },
    "company": "NEXGENE",
    "email": "audra.mendez@nexgene.io",
    "phone": "+1 (812) 461-2870",
    "address": "715 Ebony Court, Logan, New Hampshire, 9176",
    "about": "Sit deserunt nulla nulla incididunt et dolor anim eiusmod exercitation. Dolore dolor elit
```

不过我们传入的却是`..%2F..%2F..%2F..%2Fetc%2fpasswd`，最终拼接出来的文件url即为：



返回后获取到的`resource`即为`/etc/passwd`，调用`StreamUtils.copyToString(is, Charset.forName("UTF-8")`读取到文件内容：



漏洞补丁

https://github.com/spring-cloud/spring-cloud-config/commit/3632fc6f64e567286c42c5a2f1b8142bfde505c2

主要在获取到local后进行了判断：

```
if (!isInvalidPath(local) && !isInvalidEncodedPath(local)) {
   Resource file = this.resourceLoader.getResource(location)
           .createRelative(local);
   if (file.exists() && file.isReadable()) {
       return file;
   }
}
```

`isInvalidPath`用于检测其中是否含有:`/`、`..`、`WEB-INF`等关键字样，`isInvalidEncodedPath`中在进行编解码后仍是调用`isInvalidPath`进行检测。

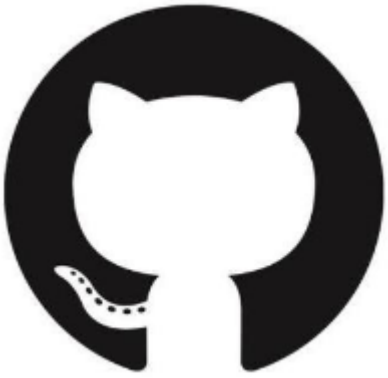
上一篇：plaidCTF两道web题目wr... 下一篇：深入分析 Windows API ...

1. 3 条回复

suolong 2019-04-19 15:23:23

感谢分享，文章中的环境搭建git地址 如果用当前版本的环境是不可以的，因为那个git地址升级了补丁所以需要找没升级补丁的版本 分享个靶机环境 https://github.com/pe4ch/cve-hub/tree/master/cve-2019-3799

1 回复Ta



chybeta 2019-04-19 20:11:48

@suolong 嗯嗯，需要checkout一下

0 回复Ta

aries 2019-04-20 09:42:45

@suolong 感谢分享。

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板