**SySeVR环境复现**

**1.实验清单**

操作系统版本： kali-linux-2022.3

软件版本：均为李珍老师在github上提供的软件：joern-0.3.1、neo4j-2.1.5、ant-1.9.14、python-joern-0.3.1、py2neo-2.0，python2和python3的版本均用kali自带，jdk用李珍老师提供的jdk1.7.0\_07.

**2.虚拟机搭建**

参考文章：[kali搭建](https://blog.csdn.net/weixin_52521462/article/details/127846626)，在官网下载kali，用virtual box的注册功能，选择蓝色小盒子图标文件。然后就可以直接打开kali，账号密码都为kali。

**1.kali换源**

可以选用：

|  |
| --- |
| JSON #中科大 deb http://mirrors.ustc.edu.cn/kali kali-rolling main non-free contrib deb-src http://mirrors.ustc.edu.cn/kali kali-rolling main non-free contrib   #阿里云 deb http://mirrors.aliyun.com/kali kali-rolling main non-free contrib deb-src http://mirrors.aliyun.com/kali kali-rolling main non-free contrib   #浙大 deb http://mirrors.zju.edu.cn/kali kali-rolling main contrib non-free deb-src http://mirrors.zju.edu.cn/kali kali-rolling main contrib non-free |

|  |
| --- |
| Plain Text sudo -i vim /etc/apt/sources.list //编辑源的文件 |

**2.查看python版本**

因为SySeVR系统需要python3和python2（joern-0.3.1的运行环境为python2.7）kali已经安装过python2和python3了。

|  |
| --- |
| JSON python2 -V Python 2.7.18  python -V Python 3.11.2 |

**3.安装JDK**

本系统所用的neo4j对应的JDK版本为1.7。

选择直接下载 jdk-7u7-linux-x64.tar.gz。

解压文件：

|  |
| --- |
| JSON sudo tar -zxvf jdk-7u7-linux-x64.tar.gz |

将jdk目录复制到/usr目录下：

|  |
| --- |
| JSON cp -r jdk1.7.0\_07 /usr  cd /usr |

修改环境变量：

|  |
| --- |
| JSON vim /etc/zsh/zshrc  #java install export JAVA\_HOME=/usr/jdk1.7.0\_07 export CLASSPATH=.:$JAVA\_HOME/lib export PATH=$PATN/bin:$PATH |

红色部分是下载的java的地址，按照自己下载的版本配置。

更新配置环境：

|  |
| --- |
| JSON source /etc/zsh/zshrc |

安装注册JDK，目的是让系统知道我们已经安装了jdk1.7.0\_07：

|  |
| --- |
| JSON update-alternatives --install /usr/bin/java java /usr/jdk1.7.0\_07/bin/java 1 update-alternatives --install /usr/bin/javac javac /usr/jdk1.7.0\_07/bin/javac 1 |

切换java版本：

|  |
| --- |
| JSON update-alternatives --config java |

手动选择1.7版本的java

跳转到./jdk1.7.0\_7/bin/目录，在本目录下查看java和javac的版本，中间需要改文件权限;

|  |
| --- |
| JSON ./java -version //查看java版本 chmod 777 java //修改java文件权限 ./javac -version  chmod 777 javac |

跳转目录查看java版本：

|  |
| --- |
| JSON java -version javac -version |

如果显示找不到java时路径可能是出现了问题，请先检查路径。

**4.安装neo4j**

将从github上下载的文件传到虚拟机上，将解压出来的SySeVR文件夹移动到/home下(保证与源码相同的路径)：

|  |
| --- |
| JSON mv SySeVR /home/ cd /home/SySeVR ls |

修改环境变量：

|  |
| --- |
| JSON vim /etc/zsh/zshrc   #install neo4j export NEO4J\_HOME=/home/SySeVR/neo4j export PATH=$PATH/bin:$PATH |

更新配置文件：

|  |
| --- |
| JSON source /etc/zsh/zshrc |

进入neo4j的conf目录，修改neo4j-server.properties：

|  |
| --- |
| JSON vim neo4j-server.properties |

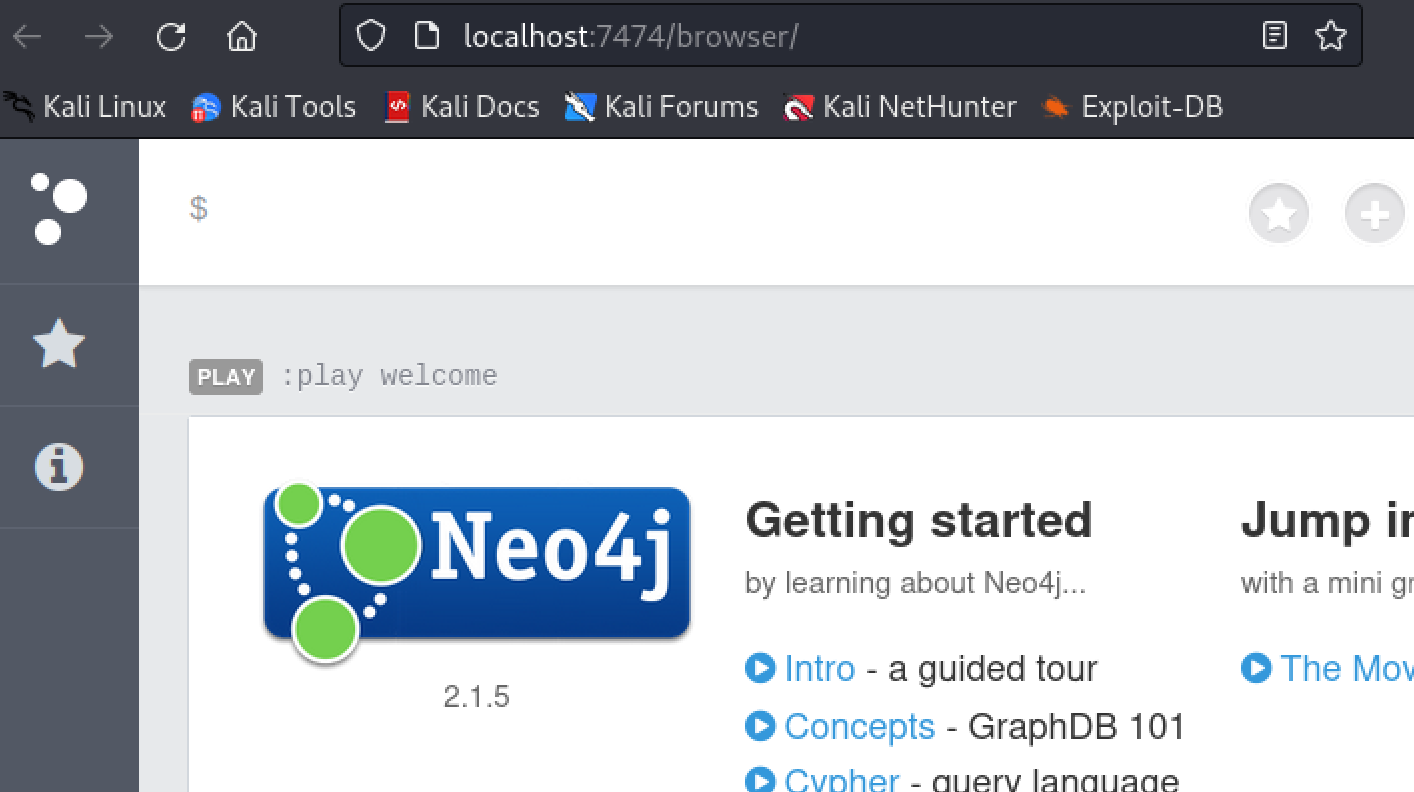
确保location指向./joernIndex文件。如果按照之前的配置路径应该是相同的。

进入bin文件夹，修改neo4j的权限，开启neo4j:

|  |
| --- |
| JSON chmod neo4j //修改权限 ./neo4j start //运行neo4j |

如果报错是可能是路径问题，检查路径。

打开链接：



**5.安装ant**

设置环境变量：

|  |
| --- |
| JSON vim /etc/zsh/zshrc   export ANT\_HOME=/home/SySeVR/softdir/apache-ant-1.9.14 expoer PATH=$ANT\_HOME/bin:$PATH |

更新环境：

|  |
| --- |
| JSON source /etc/zsh/zshrc |

查看ant版本：

|  |
| --- |
| JSON ant -version |

**6.安装joern**

设置环境变量并更新：

|  |
| --- |
| JSON vim /etc/zsh/zshrc  export JOERN\_HOME=/home/SySeVR/joern-0.3.1 source /etc/zsh/zshrc |

在joern-0.3.1文件下，输入"ant" --我的理解是，建立ant与joern的联系:

|  |
| --- |
| JSON cd /home/SySeVR/joern-0.3.1 ant |

在用一个目录下，输入"ant tools":

|  |
| --- |
| JSON ant tools |

**7.py2neo安装**

|  |
| --- |
| JSON cd /home/SySeVR/softdir/py2neo-py2neo-2.0 python2 setup.py install |

**8.安装python-joern-0.3.1**

|  |
| --- |
| JSON cd /home/SySeVR/softdir/python-joern-0.3.1 python2 setup.py install |

**9.验证joern是否安装成功**

因为新版的压缩包里，没joern-tool这个安装包，因此再joern-tools目录下用joern-lookup不可行。

查看joern目录下的全部文件：

|  |
| --- |
| JSON ls -all |

删除.joernIndex文件，因为你每次进行新的实验，都需要将该文件删除，这个文件是joern与neo4j之间的"桥梁"。

|  |
| --- |
| JSON rm -rf .joernIndex |

将测试数据导入joern:

|  |
| --- |
| JSON java -jar /home/SySeVR/joern-0.3.1/bin/joern.jar /home/SySeVR/testcode |

参考文章：

[SySeVR环境配置](https://blog.csdn.net/qq_55202378/article/details/127452821)，[SySeVR环境配置下](https://blog.csdn.net/qq_55202378/article/details/127454191)

**3.测试**

使用github上文件本身提供的测试样例。

**将测试数据导入joern：**

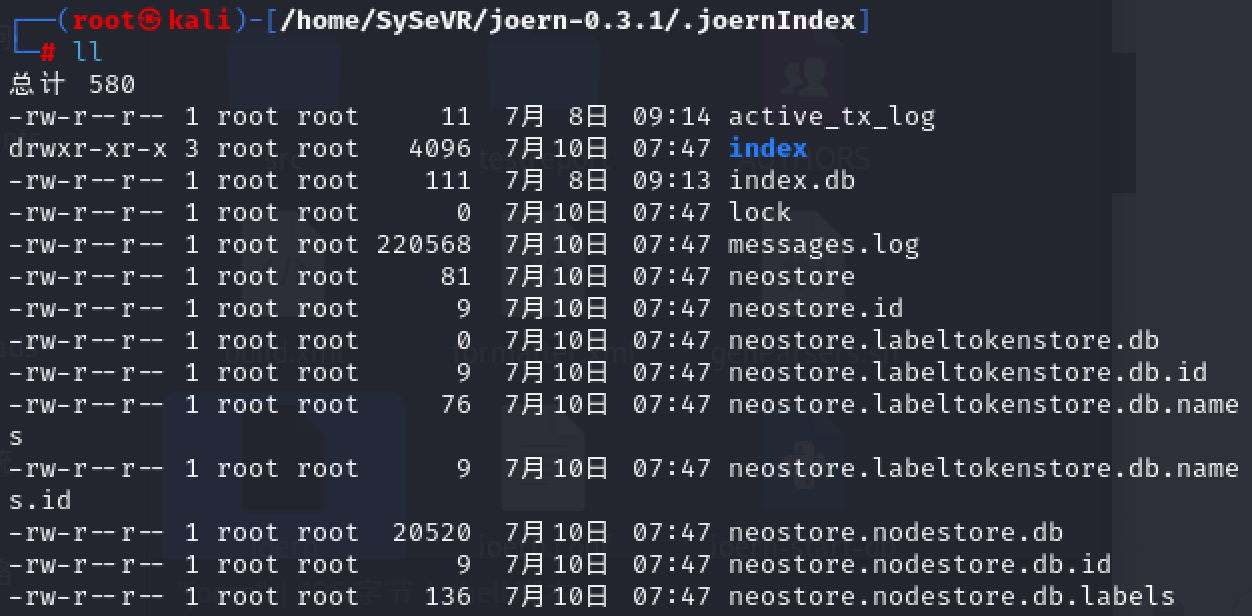
在/home/SySeVR/joern-0.3.1查看是否存在.joernIndex文件，有的话，需要删除。

|  |
| --- |
| JSON rm -rf .joernIndex |

删除之后，将测试数据导入joern：

|  |
| --- |
| JSON java -jar /home/SySeVR/joern-0.3.1/bin/joern.jar /home/test/testCode |

查看.JoernIndex



**打开neo4j**

|  |
| --- |
| JSON neo4j start |

**开始切片**

**1、get\_cfg\_relation.py**

|  |
| --- |
| JSON cd source2slice python2 get\_cfg\_relation.py |

**bug1：报错显示找不到igraph库**

|  |
| --- |
| JSON pip2 install python-igraph==0.7.1.post6 |

报错显示没有c core

**下载igraph-0.7.1**

在https://github.com/igraph/igraph/releases?page=3下载igraph-0.7.1.tar.gz

解压缩：

|  |
| --- |
| JSON tar -zxvf igraph-0.7.1.tar.gz |

安装：

|  |
| --- |
| JSON cd igraph-0.7.1 ./configure make make install |

重新下载python-igraph

|  |
| --- |
| JSON pip2 install -v python-igraph==0.7.1.post6 --install-option="--c-core-url=home/kali/igraph-0.7.1.tar.gz" |

**bug2:python 无法找到 libigraph.so.0 这个共享库文件**

1. 找到 libigraph.so.0 这个文件

|  |
| --- |
| Bash sudo find / -name libigraph.so.0 |

1. 通过命令将其添加到库路径中

|  |
| --- |
| Bash echo "/usr/local/lib" | sudo tee -a /etc/ld.so.conf  #将 /usr/local/lib 添加到 /etc/ld.so.conf 文件的末尾 sudo ldconfig #更新库路径 |

1. 重新运行脚本

**bug3：没有‘cfg\_db/testCode’目录。**

|  |
| --- |
| JSON mkdir cfg\_db |

**2、complete\_PDG.py**

|  |
| --- |
| JSON python2 complete\_PDG.py |

**bug1：没有pdg\_db/testCode文件夹**

解决方法：

|  |
| --- |
| JSON mkdir pdg\_db |

**3、access\_db\_operate.py**

|  |
| --- |
| JSON python2 access\_db\_operate.py |

**bug1：报错：没有dict\_call2cfgNodeID\_funcID/testCode文件夹**

解决方法：

|  |
| --- |
| JSON mkdir dict\_call2cfgNodeID\_funcID |

**4、points\_get.py**

|  |
| --- |
| JSON python2 points\_get.py |

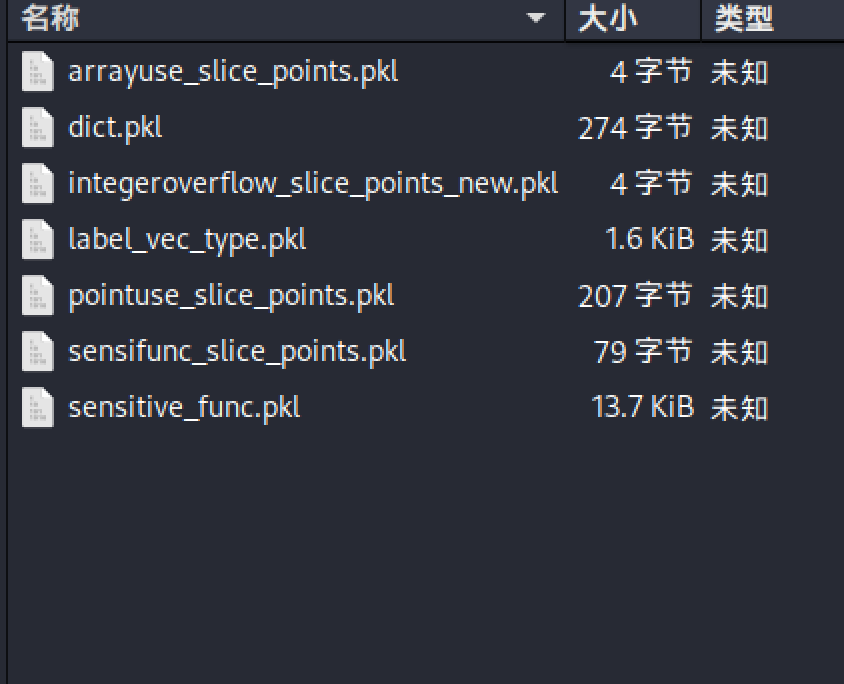
这一步输出文件在source2slice目录下，有四个输出文件，分别为：

sensifunc\_slice\_points.pkl

pointuse\_slice\_points.pkl

arrayuse\_slice\_points.pkl

integeroverflow\_slice\_points\_new.pkl



**5、extract\_df.py**

|  |
| --- |
| JSON python2 extract\_df.py |

**bug1：No such file or directory: C/test\_data/4/api\_slices.txt**

解决方法：

|  |
| --- |
| JSON  mkdir -pv C/test\_data/4 |

在目标文件夹里生成了两个txt文件：

