

源码

简介

这是“知云破晓——云服务软件供应链治理系统”中软件成分分析子系统的工具geekloud工具的源码。除此之外，包含了测试代码。

本工具使用Python语言，请在Linux服务器下构建和使用。

结构

结构如下：

```
.
├── analysis_tools.py
├── builder.py
├── cydx2geekloud.py
├── database.py
├── dep-reports #组件SBOM输出报告
├── entry.py
├── geekloud2cydx.py
├── geekloud2spdx.py
├── geek_transfer.py
├── getattr.gradle
├── __init__.py
├── interface.py
├── interface.spec #geekloud打包配置文件
├── logs #日志目录
├── output #输出结果
├── __pycache__
├── repo #测试组件仓库
├── repo.py
├── SBOM #SBOM库
├── setup_env.py
├── software_jsonfiles #组件知识库
├── spdx2geekloud.py
├── start_analysis.sh
├── task #组件库
├── test.py
├── tools #工具库
└── utils.py
```

代码说明

1. interface.py: 工具启动器，可以使用上述的命令直接对工具进行测试。其中有两个主要函数，`parse_args` 用来处理命令，`dependency` 用来启动依赖分析模块，`compare`用来启动比较分析模块，`transfer`用来启动转换分析模块。
2. builder.py: 依赖分析过程中，首先会根据包管理器类型（Maven/Gradle）获取对应的 `builder`，其中最主要的函数方法是 `parse_dependency`，对目标组件进行依赖分析，并输出指定格式的结果。除此之外，还有一些其他的工具函数，用来处理异常和获取结果。

3. analysis_tools.py: 用于比较两个SBOM的差异。
4. geek_transfer.py: geekcloud两种SBOM规范的转换。
5. 一键转换功能相关
 1. geekcloud2cydx.py
 2. geekcloud2spdx.py
 3. spdx2geekcloud.py
 4. cydx2geekcloud.py
6. setup_env.py: 主要用于环境的配置。
7. database.py: 用于对分析结果的输出, 其中有包括数据库、字典在内的三种格式, 目前使用的是其中的 dict 格式, 会将依赖分析结果解析为字典嵌套的形式。
8. utils.py: 对于 log 的一些设置, 以及启动器需要用到的一些工具函数。
9. repo.py: 用来对指定组件的 github 仓库进行下载, 可以在代码中修改对应组件。
10. entry.py: 批量化的对一个组件的所有版本进行下载和依赖分析, 之前用来测试工具功能。
- 11.

运行环境

依旧在Linux操作系统运行。

环境配置同geekcloud工具。如需, 请在 setup_env.py 中改变jdk、maven、gradle对应的环境路径。

运行方式

使用方式同geekcloud工具, 采用命令行输入的形式, 但是需要将 ./geekcloud 替换成 python ./interface.py (注意切换到代码目录下)。

example:

```
python ./interface.py --action dependency --taskid ./task/dubbo-3.2/
```

具体细节参考geekcloud工具手册。