SWRhapsody

Java 反编译工具

于1月 20, 2020由**SWRhapsody**发布

在复现 CVE-2019-15012 遇到了一个非常坑的地方，使用 Jd-gui 和 Jd-cli 反编译出来的结果与实际的代码有很大的出入。这两款工具会导致分析补丁的结果出现很大的误差。具体为什么以及什么情况下会这样还不清楚。

上面两张图中左边的是Idea 对 bitbucket-git-6.8.2.jar 的反编译，右边是Jd-gui 1.6.3 和 Jd-cli 的解析结果，当前最新版本1.6.6 也存在同样的问题。

这里推荐使用Idea 中的反编译工具（https://github.com/JetBrains/intellij-community/tree/master/plugins/java-decompiler/engine）对 Jar 进行反编译，以免出现不必要的麻烦。

顺便提供一个linux 下的批量解析脚本 auto_decompile.sh

```
#!/bin/bash
OUTDIR=$2
INDIR=$1
LIMIT=0

OPTS=""
#unpack all jar file
find $INDIR -maxdepth 20 -type f -print0 |while IFS= read -r -d $'\0' file; do
```

SWRhapsody

```
13                  RELFILE=$(realpath --relative-to=$INDIR $file)
14                  RELPATH=${RELFILE::-4}
15
16                  mkdir -p $(dirname $OUTDIR/$RELPATH)
17                  #decompile them to the output + relative path
18                  unzip  $OUTDIR/$(basename $RELFILE) -d $OUTDIR/$RELPATH
19                  #delete the jar file in the output folder
20                  rm -r $OUTDIR/$(basename $RELFILE)
21          fi
22  done
```

解压 decompile.zip 进入目录运行 ./auto_decompile.sh <jar-dir> <output-dir>，想要忽略的库文件放入 ignore 文件夹，Fernflower 的其他用法详情请看GitHub上的Readme。

PS: 在写这个脚本的地方有一个坑，如果让 fernflower 自己遍历 jar 包，在 jar 包数量很多的时候 fernflower 会直接卡住，就算设置了超时时间也救不了。

分类: EXERCISE

0 条评论

发表评论

名称 *

电子邮件 *

网站

在想些什么?

SWRhapsody

发表评论

近期文章

- 携程Apollo YAML 反序列化
- CVE-2020-5410
- CodeQL部分源码简读
- 服务器与网关的不一致
- CodeQL 部分使用记录

近期评论

文章归档

- 2020年8月
- 2020年6月
- 2020年5月
- 2020年3月
- 2020年1月
- 2019年12月
- 2019年11月

SWRhapsody

2019年5月

2019年4月

2019年1月

2018年11月

2018年10月

2018年9月

2018年4月

2018年3月

2018年2月

2018年1月

分类目录

Article Collection

Cheat Sheet

cryptography

Exercise

Exploit

HackTheBox

Penetration Test

Uncategorized

SWRhapsody

EXERCISE

CodeQL部分源码简读

Introduction CodeQL 也用了不少时间了，从最初的不会 阅读更多…

CHEAT SHEET

CodeQL 部分使用记录

前言 CodeQL 是一个代码分析引擎，主要原理是通过对代码进行构建并 阅读更多…

EXERCISE

OpenRASP

前言 记得我还没想过要从事安全行业时看过一个黑客的博客，博主说他找到了 阅读更多…

ABOUT

Hestia |由ThemeIsle开发