基于海豚的软件服务--代码安全审核方法

# 检查端口清单

每个对外端口是否合理，是否在表格中逐一进行了说明。特别80/443以外的端口。模板中的2500/5510等字样是否已经删除。

对于myslq、es等第三方库，如果在送测软件的安装包中包含了，则也需要进行审查其端口开发情况。

# 检查服务清单

每个服务是否都提供了matadata.json文件，并且文件内容正确，其中包含了映射关系（route字段、gatewayRoutes字段等）。

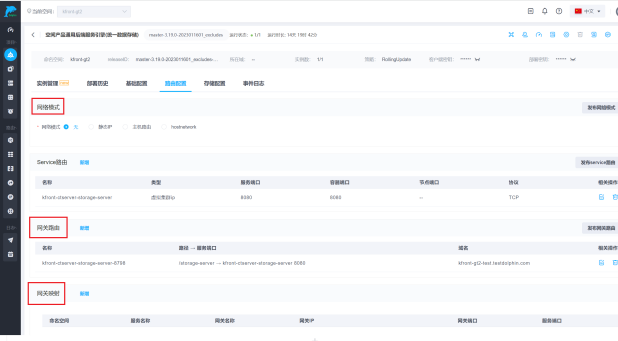
对所有服务的matadata.json以及部署脚本中的XX-values.yaml文件，进行如下检查：

gatewayRoutes：URL映射关系是否需要对外。重点审查如果是内部通信使用的URL，不应该出现。

gatewayMappings（端口映射）和hostRoutes（端口直接对外）。如果存在端口映射，检查是否在端口清单中有说明。

并且在海豚运维系统中，检查该服务的配置，是否和提供的matadata.json中的内容一致。

matadata.json文件的示例：



# 对每个对外访问的服务进行检查

对于不提供外部访问的服务（URL映射、端口映射均不存在），无需进行后续服务检查。

通过海豚运维系统，逐一登录每个对外提供服务的容器终端界面中

通过一下命令检查后台进程，区分不同的服务类型：

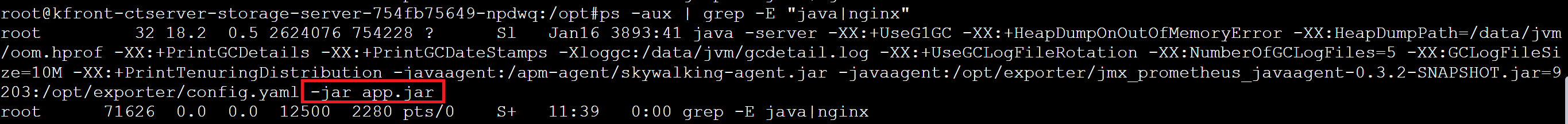
ps -aux | grep -E 'java|nginx'

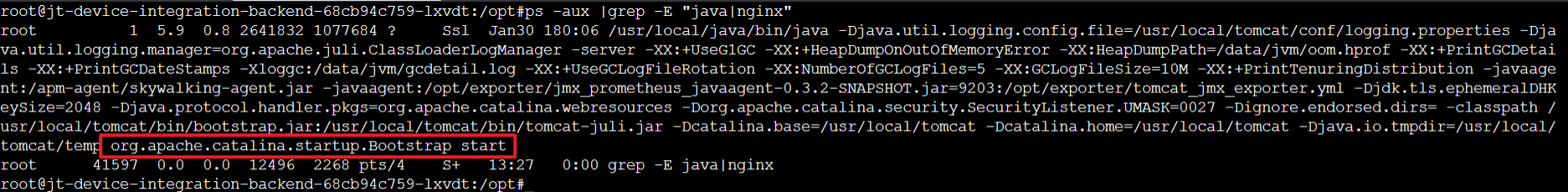
如果存在-ar app.jar等程序，一般是存在springboot单体web程序。

如果存在tomcat或catalina关键字，是存在tonmcat服务。

如果存在nginx等程序，一般是存在web前端服务。

例子：





按照工程类型做不同检查：（同时存在多个服务类型，按不同类型均做检查）

## 前端服务

前端一般是没有登录鉴权的，所有的数据都认为是可以任意访问的。在Web服务中查找是否有敏感信息泄露，Token泄露和密码泄露

一般Web主目录是/app\_home目录，在该目录下搜索是否存在有问题的关键字。

*cd /app\_home*

*grep -i -E "token|password|passwd|pwd|secret" \*/\*.json  
grep -i -E "token|password|passwd|pwd|secret" \*/\*/\*.json  
grep -i -E "token|password|passwd|pwd|secret" \*/\*/\*/\*.json  
grep -i -E "token|password|passwd|pwd|secret" \*/\*/\*/\*/\*.json  
grep -i -E "token|password|passwd|pwd|secret" \*/\*/\*/\*/\*.json  
grep -i -E "token|password|passwd|pwd|secret" \*/\*/\*/\*/\*/\*.json  
grep -i -E "token|password|passwd|pwd|secret" \*/\*/\*/\*/\*/\*/\*.json*

在搜索出来的内容中，检查是否存在默认token、密码信息泄露等问题。

## 后端服务tomcat服务或springboot单体服务

后端主要检查是否有鉴权，由于我司大部分服务为springboot程序，可以通过检查application.yml、application-online.yml、application-dev.yml等文件，来检查是否正确配置了rbac拦截器。

如果是springboot单体服务，需要先解压缩ps -aux中看到的app.jar文件，一般该文件在/app\_home/app目录下，执行解压缩

*cd /app\_home/app*

*mkdir tmp*

*cp app.jar tmp/*

*cd tmp*

*jar xf app.jar*

*（检查结束记得删除tmp文件夹）*

如果是tomcat服务，直接cd到tomcat所在目录下的webapps目录

*cd /usr/local/tomcat/webapps*

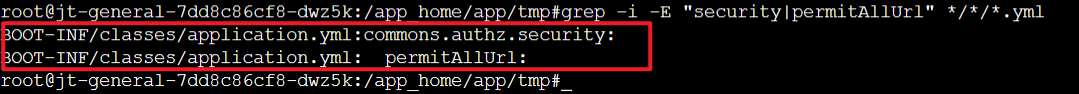
可以通过命令进行全文搜索,来检查是否正确配置了rbac拦截器。

*grep -i -E "security|permitAllUrl" \*/\*.yml*

*grep -i -E "security|permitAllUrl" \*/\*/\*.yml*

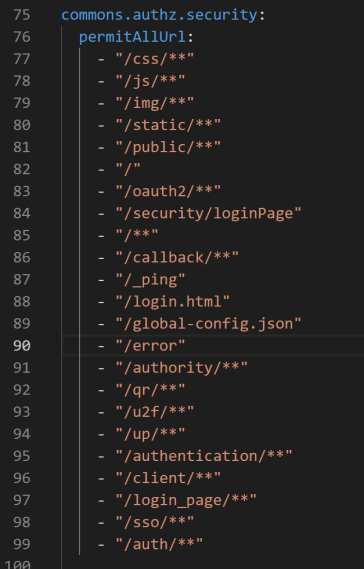
*grep -i -E "security|permitAllUrl" \*/\*/\*/\*.yml*

如果使用了rbac拦截器，应该可以查到以下信息：



如果没有查询到以上信息，表示未使用RBAC进行鉴权拦截检查或者采用了其他方式。需要和申请人进行沟通。

如果查询到以上信息，需要打开该yml文件，检查相关配置.



其中permitAllUrl表示不需要鉴权就可以直接访问的URL，需要检查该列表是否合理。特别关注不应该存在的URL比如：/ws、/\*\*等不应该出现。

对于如果/download等URL无法明确这些URL是否可以公开的，需要在《代码审查表》中对这些放行不检查的URL予以说明原因，并审查其合理性。

对application.yml文件中的permiAllUrl字段进行抽检，检查和《代码审查表》中嵌入的application.yml等附件是否一致。

# 对其他检查项目检查【通用】

一般来说，前后端分离的系统（前端使用nginx，后端使用springboot），很少出现以下审查项的问题。【WebServer使用不当漏洞】【WebServer配置不当漏洞】

需要检查【WebServer任意文件下载漏洞】【业务程序任意文件下载漏洞】、【业务程序任意文件上传漏洞】，重点关注业务功能上的下载等功能，由哪些服务提供。顺着业务功能的设计方案进行审查。

对于【SQL注入漏洞】【脚本注入漏洞】，关注提供的截图是否正确。

对于【私有协议安全】，仅关注是否进行了有效的鉴权。需要在《审查表》中，说明清楚了鉴权机制。