

Checkmarx activities – Conduct Backend

Query Name: Improper_Resource_Access_Authorization

Qty: 150 – (37 not exploitable)

Cause: most of them are about sql query and those ones are not exploitable were put with a comment explaining that it is used JPQL with named parameters instead of standard sql language, so I think it is the same for the rest that is similar. Ex.: id 154 and 155 (not exploitable); 245 and 246 (not exploitable).

Query Name: Spring_Missing_Object_Level_Authorization and Spring_Missing_Function_Level_Authorization

Qty: 76

```
50
51
52 @PostMapping(path = "/miniSuper/affect")
53 public void definePerimetersOfMiniSuper(@RequestBody MiniSuperPerimetersDto repositories) {
54     this.accountService.addPerimetersToMiniSuper(repositories.getPerimetersToAdd());
55     this.accountService.removePerimetersFromMiniSuper(repositories.getPerimetersToRemove());
56 }
57
58 public Boolean notAuthorized(final UserDTO userDTO) {
59     return userDTO.isAuthenticated() && Objects.nonNull(userDTO.getUser()) && !userDTO.getUser().g
```

Line 52 flags a method or annotation that could be a potential unauthorized access to object available in the corresponding controller. This query is looking for possible flaws in Spring-Security configuration, so only projects that use Spring-Security are considered.

Result State	Graph	Codebashing	Id	Direct	Query Name	Status	Source Folder	Source Filename	Source	Source Object	Destination F
<input type="checkbox"/>					spring_missing	All					
<input type="checkbox"/>			452		Spring_Missing_Content_Security_Policy	Recurr...	\src\main\java\...	MISAdapter.java	10	annotation	\src\main\j
<input type="checkbox"/>			453		Spring_Missing_Expect_CT_Header	Recurr...	\src\main\java\...	MISAdapter.java	10	annotation	\src\main\j
<input checked="" type="checkbox"/>			335		Spring_Missing_Function_Level_Authorization	Recurr...	\src\main\java\...	AccountControll...	51	PostMapping	\src\main\j
<input type="checkbox"/>			336		Spring_Missing_Function_Level_Authorization	Recurr...	\src\main\java\...	FileController.ja...	60	PostMapping	\src\main\j
<input type="checkbox"/>			337		Spring_Missing_Function_Level_Authorization	Recurr...	\src\main\java\...	FileController.ja...	106	DeleteMapping	\src\main\j
<input type="checkbox"/>			338		Spring_Missing_Function_Level_Authorization	Recurr...	\src\main\java\...	FileController.ja...	120	PutMapping	\src\main\j
<input type="checkbox"/>			339		Spring_Missing_Function_Level_Authorization	Recurr...	\src\main\java\...	IndicatorControl...	33	PostMapping	\src\main\j

Cause: GetMapping annotation. Solutions in the internet point to add the annotation PreAuthorize to check if it is authenticated. Ex.: id 295 and 335.

Query Name: Spring_Overly_Permissive_Cross-Origin_Resource_Sharing_Policy

Qty: 78

```
33
34
35 @GetMapping(path = "/constants")
36 public List<MisRef> getTranslationsByParent(@RequestParam(value="parent") String parent) {
37     return this.translateService.findByParent(parent);
38 }
39
```

The method getTranslationsByParent found at line 35 in src\main\java\com\natixis\mis\controller\TranslateController.java sets an overly permissive CORS access control origin header.

Result State	Graph	Codebashing	Id	Direct	Query Name	Status	Source Folder	Source Filename	Source	Source Object	Destination Folder
<input type="checkbox"/>					Spring_Overly_Permissive_Cross-Origin_R	All					
<input type="checkbox"/>			446		Spring_Overly_Permissive_Cross-Origin_Resou...	Recurr...	\src\main\java\...	TranslateContro...	25	getTranslations	\src\main\java\...
<input type="checkbox"/>			447		Spring_Overly_Permissive_Cross-Origin_Resou...	Recurr...	\src\main\java\...	TranslateContro...	30	getAllConstants	\src\main\java\...
<input checked="" type="checkbox"/>			448		Spring_Overly_Permissive_Cross-Origin_Resou...	Recurr...	\src\main\java\...	TranslateContro...	35	getTranslations...	\src\main\java\...
<input type="checkbox"/>			449		Spring_Overly_Permissive_Cross-Origin_Resou...	Recurr...	\src\main\java\...	UserController.j...	91	getUser	\src\main\java\...
<input type="checkbox"/>			450		Spring_Overly_Permissive_Cross-Origin_Resou...	Recurr...	\src\main\java\...	UserController.j...	38	getContext	\src\main\java\...

Cause: They are composed by methods in which are annotated with PostMapping and GetMapping. Solutions in the internet point to add white/black list and controlling better instead of using wildcard. Ex.: 448 and 449 (not exploitable – getUser method validates if the user is logged in, if not it redirects to the SSO login page)