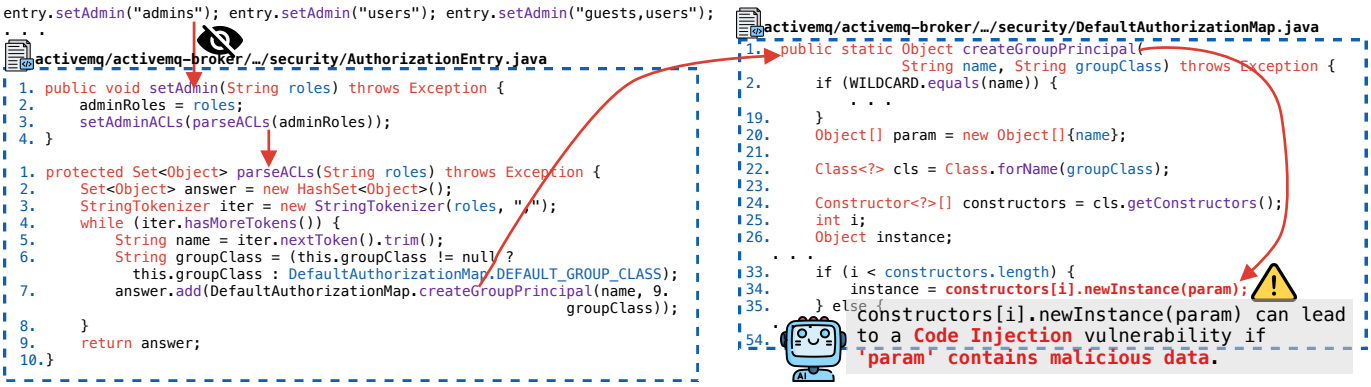

Fig. 8: An illustrating example FP reported by IRIS of reason A1

APPENDIX A
CASE STUDY

In RQ3 (Section V-C), we introduced a taxonomy of false-positive (FP) causes detected by the selected methods and showcased two representative examples for categories A2 and B1. In this section, we present the remaining examples of all 9 categories.

1) Example Program for Reason A1: Figure 8 illustrates an FP reported by IRIS in the `apache/activemq` project. IRIS (via the LLM) flags the creation of a new instance with `param` as a potential code-injection risk, assuming that `param` may contain attacker-controlled content. However, tracing the data flow shows that `param` originates from the `roles` argument of `setAdmin`, is propagated through `parseACLs`, and is ultimately passed into `createGroupPrincipal`. Further inspection of all call sites of `setAdmin` in the project indicates that the supplied arguments are hard-coded string literals (e.g., `"admins"`, `"users"`, ...), and therefore cannot contain malicious input. This FP arises because the LLM's analysis considers the data flow starting at `roles` but does not incorporate the concrete invocations of `setAdmin`, leading it to incorrectly treat `param` as untrusted.

2) Example Program for Reason C1: Figure 9 is an FP due to the prompt deviations (reason C1). As shown in the prompt, IRIS explicitly asks the LLM to determine whether the dataflow contains a `CWE-022` vulnerability, and provides the corresponding dataflow details. Based on this information, the LLM can infer that the flow starts from reading an attribute value from an XML element and then uses it as a parameter in a database query. The LLM correctly recognizes that this pattern is unrelated to `CWE-022` (Path Traversal or Zip Slip). Nevertheless, it still labels the code as vulnerable, despite being explicitly instructed to judge only whether the provided content constitutes a `CWE-022` issue.

3) Example Program for Reason D1: Figure 10 illustrates an FP reported by RepoAudit caused by missing a key program point (Reason D1). In this example, `req` may be assigned `NULL` by `peci_request_alloc`. RepoAudit infers that this `req` is subsequently passed to `peci_request_xfer_retry`, where the null pointer is dereferenced via `req->device`, and therefore reports a potential null pointer dereference (NPD). However, the tool

Fig. 9: An illustrating example FP reported by IRIS of reason C1

overlooks a simple but crucial null-check along this path: when `req` is `NULL`, the function returns `ERR_PTR` at line 12, which prevents the dereference. Notably, all three related functions shown in this figure are located in the same file and involve neither complex nor long contexts, yet RepoAudit still fails due to this reason.

4) Example Program for Reason D2: There is an FP case reported by INFERROI due to limited support for Java language features (Reason D2). In this

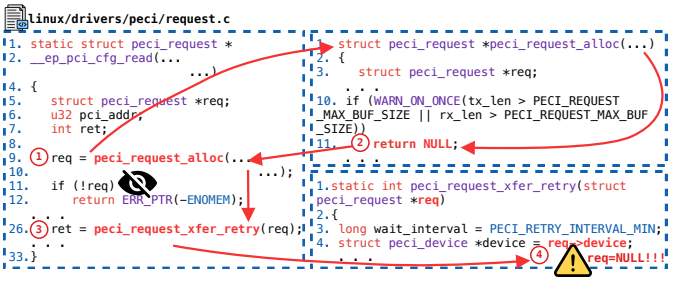

Fig. 10: An illustrating example FP reported by RepoAudit of reason D1

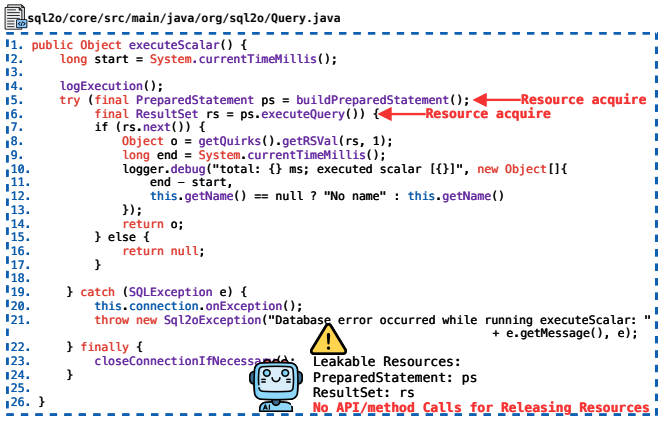

Fig. 11: An illustrating example FP reported by INFERROI of reason D2

example, the LLM correctly identifies two resource-acquisition operations, `buildPreparedStatement()` and `ps.executeQuery()`. However, it overlooks that both are declared in a `try-with-resources` statement and thinks there are no API/method calls for resource release. When the block exits, the corresponding `PreparedStatement` and `ResultSet` are automatically closed, so no resource leak occurs.

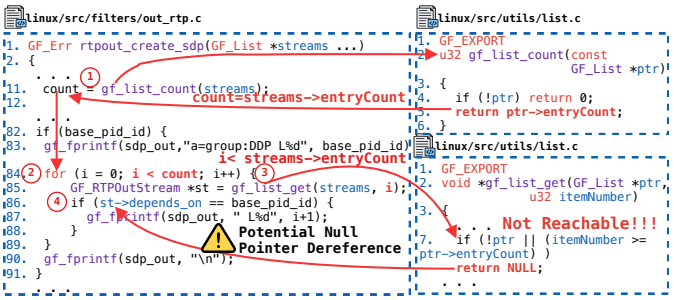

Fig. 12: An illustrating example FP reported by CodeQL of reason D3

5) Example Program for Reason D3: Figure 12 presents an FP reported by CodeQL caused by path-insensitive analysis (Reason D3). As shown, CodeQL thinks that the call to `gf_list_get` may return `NULL` and assigns it to `st` (Line 85), after which `st` is dereferenced via `st->depends_on` (Line 86). It therefore flags a potential null-pointer dereference (NPD). However, in this context, `count` is initialized with `streams->entryCount` (Line 11), which ensures that the `return NULL;` branch in `gf_list_get` is unreachable. Consequently, `st` cannot be `NULL` at Line 86, and the reported NPD is a false alarm.

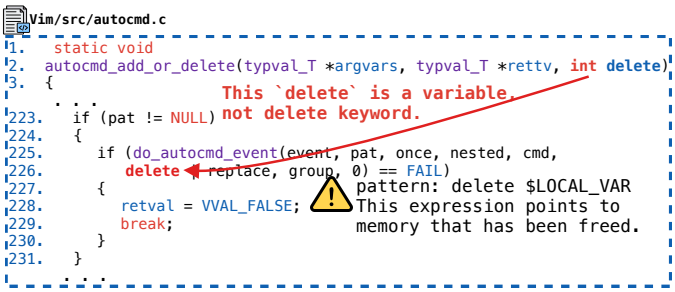

Fig. 13: An illustrating example FP reported by Semgrep of reason D4

6) Example Program for Reason D4: There is an FP reported by Semgrep caused by missing type awareness (Reason D4). In this example, Semgrep flags a `delete` operation at Line 226 as if it were freeing allocated memory and then being used in a subsequent function call. However, `delete` here is an integer parameter rather than the C++ memory-deallocation operator, and thus it does not perform any memory-related operation. As a result, the reported issue is a false alarm.

APPENDIX B
CONFIGURATION OF TRADITIONAL STATIC METHODS

To evaluate CodeQL and Semgrep on the two datasets used in this paper, we manually selected from their built-in rule sets those checks that are related to the CWE types under study. Table VIII and Table IX demonstrate the details rules selected for each CWE type. Although different rules may produce reports with different severity levels (e.g., some are marked as errors and some are only recommendations), we treat them equally in this study.

APPENDIX C
DETAILED OVERHEAD OF SELECTED METHODS

In Section V-D, we report the average overhead of the evaluated methods on the real-world projects. Here, we provide the complete per-project overhead details for each method in Table X and Table XI.

TABLE VIII: Selected CodeQL rules for each CWE type

CodeQL	CWE-401	Critical/FileMayNotBeClosed.ql, Critical/FileNeverClosed.ql, Critical/MemoryMayNotBeFreed.ql, Critical/MemoryNeverFreed.ql Critical/NewArrayDeleteMismatch.ql, Critical/DescriptorMayNotBeClosed.ql, CWE-401/MemoryLeakOnFailedCallToRealloc.ql
	CWE-416	Critical/DoubleFree.ql, Critical/UseAfterFree.ql, Critical/ReturnStackAllocatedObject.ql CWE-416/IteratorToExpiredContainer.ql, CWE-416/UseOfStringAfterLifetimeEnds.ql, CWE-416/UseOfUniquePointerAfterLifetimeEnds.ql
	CWE-476	Critical/InconsistentNullnessTesting.ql, Critical/MissingNullTest.ql, Critical/NotInitialised.ql CWE-476/DangerousUseOfExceptionBlocks.ql, Critical/GlobalUseBeforeInit.ql
	CWE-022	CWE-022/TaintedPath.ql, CWE-022/ZipSlip.ql
	CWE-078	CWE-078/ExecRelative.ql, CWE-078/ExecTainted.ql, CWE-078/ExecTaintedEnvironment.ql CWE-078/ExecUnescaped.ql, experimental/CWE-078/ExecTainted.ql, experimental/CWE-078/CommandInjectionRuntimeExec.ql experimental/CWE-078/CommandInjectionRuntimeExecLocal.ql
	CWE-079	CWE-079/AndroidWebViewAddJavascriptInterface.ql, CWE-079/AndroidWebViewSettingsEnabledJavaScript.ql, CWE-079/XSS.ql
	CWE-094	CWE-094/ArbitraryApkInstallation.ql, CWE-094/GroovyInjection.ql, CWE-094/InsecureBeanValidation.ql, CWE-094/JexlInjection.ql CWE-094/MvelInjection.ql, CWE-094/SpelInjection.ql, CWE-094/TemplateInjection.ql, experimental/CWE-094/BeanShellInjection.ql experimental/CWE-094/InsecureDexLoading.ql, experimental/CWE-094/JakartaExpressionInjection.ql experimental/CWE-094/JShellInjection.ql, experimental/CWE-094/JythonInjection.ql, experimental/CWE-094/SpringViewManipulation.ql experimental/CWE-094/ScriptInjection.ql, experimental/CWE-094/SpringImplicitViewManipulation.ql
	CWE-722	Resource Leaks/CloseReader.ql, Resource Leaks/CloseSql.ql, Resource Leaks/CloseWriter.ql

TABLE IX: Selected Semgrep rules for each CWE type

Semgrep	CWE-401	semgrep-rules/c/mismatched-memory-management-cpp.yaml, semgrep-rules/c/mismatched-memory-management.yaml
	CWE-416	r/cpp.lang.security.containers.std-vector-invalidation.std-vector-invalidation r/cpp.lang.security.strings.return-c-str.return-c-str, r/c.lang.security.use-after-free.use-after-free r/cpp.lang.security.strings.string-view-temporary-string.string-view-temporary-string r/cpp.lang.security.use-after-free.local-variable-malloc-free.local-variable-malloc-free r/cpp.lang.security.use-after-free.local-variable-new-delete.local-variable-new-delete r/c.lang.security.function-use-after-free.function-use-after-free
	CWE-476	r/cpp.lang.security.memory.null-deref.null-library-function.null-library-function
	CWE-022	r/java.jax-rs.security.jax-rs-path-traversal.jax-rs-path-traversal, r/java.lang.security.httpservlet-path-traversal.httpservlet-path-traversal r/java.micronaut.path-traversal.file-access-taint-msg.file-access-taint-msg, r/java.micronaut.path-traversal.file-access-taint-sls.file-access-taint-sls r/java.micronaut.path-traversal.file-access-taint-ws.file-access-taint-ws, r/java.micronaut.path-traversal.file-access-taint.file-access-taint r/java.micronaut.path-traversal.file-taint-msg.file-taint-msg, r/java.micronaut.path-traversal.file-taint-sls.file-taint-sls r/java.micronaut.path-traversal.file-taint-ws.file-taint-ws, r/java.micronaut.path-traversal.file-taint.file-taint r/java.servlets.security.httpservlet-path-traversal-deepsemgrep.httpservlet-path-traversal-deepsemgrep r/java.servlets.security.httpservlet-path-traversal.httpservlet-path-traversal r/java.spring.spring-tainted-path-traversal.spring-tainted-path-traversal r/gitlab.find_sec_bugs.FILE_UPLOAD_FILENAME-1, r/gitlab.find_sec_bugs.PATH_TRAVERSAL_IN-1 r/gitlab.find_sec_bugs.PATH_TRAVERSAL_OUT-1.PATH_TRAVERSAL_OUT-1, r/gitlab.find_sec_bugs.PT_ABSOLUTE_PATH_TRAVERSAL-1 r/gitlab.find_sec_bugs.PT_RELATIVE_PATH_TRAVERSAL-1, r/gitlab.find_sec_bugs.WEAK_FILENAMEUTILS-1
	CWE-078	r/java.lang.security.audit.command-injection-formatted-runtime-call.command-injection-formatted-runtime-call r/java.lang.security.audit.command-injection-process-builder.command-injection-process-builder r/java.micronaut.command-injection.tainted-system-command-msg.tainted-system-command-msg r/java.micronaut.command-injection.tainted-system-command-sls.tainted-system-command-sls, r/java.micronaut.command-injection.tainted-system-command-ws.tainted-system-command-ws r/java.micronaut.command-injection.tainted-system-command.tainted-system-command r/java.servlets.security.tainted-cmd-from-http-request-deepsemgrep.tainted-cmd-from-http-request-deepsemgrep r/java.servlets.security.tainted-cmd-from-http-request.tainted-cmd-from-http-request java.spring.command-injection.tainted-system-command.tainted-system-command r/java.spring.simple-command-injection-direct-input.simple-command-injection-direct-input r/java.lang.security.audit.tainted-cmd-from-http-request.tainted-cmd-from-http-request r/java.spring.security.injection.tainted-system-command.tainted-system-command r/gitlab.find_sec_bugs.COMMAND_INJECTION-1, r/mobsf.mobsfscan.injection.command_injection.command_injection r/mobsf.mobsfscan.injection.command_injection_formated.command_injection_warning
	CWE-079	java.lang.security.audit.xss.no-direct-response-writer.no-direct-response-writer java.lang.security.servletresponse-writer-xss.servletresponse-writer-xss, java.micronaut.xss.direct-response-write.direct-response-write java.servlets.security.no-direct-response-writer-deepsemgrep.no-direct-response-writer-deepsemgrep java.servlets.security.no-direct-response-writer.no-direct-response-writer, java.micronaut.xss.direct-response-write.direct-response-write java.servlets.security.servletresponse-writer-xss-deepsemgrep.servletresponse-writer-xss-deepsemgrep java.servlets.security.servletresponse-writer-xss.servletresponse-writer-xss java.spring.tainted-html-string-responsebody.tainted-html-string-responsebody
	CWE-094	r/gitlab.find_sec_bugs.TEMPLATE_INJECTION_PEBBLE-1.TEMPLATE_INJECTION_FREEMARKER-1.TEMPLATE_INJECTION_VELOCITY-1, r/gitlab.find_sec_bugs.SCRIPT_ENGINE_INJECTION-1.SPEL_INJECTION-1.EL_INJECTION-2.SEAM_LOG_INJECTION-1 r/java.lang.security.audit.el-injection.el-injection, r/java.lang.security.audit.ognl-injection.ognl-injection r/java.lang.security.audit.script-engine-injection.script-engine-injection, r/java.spring.security.audit.spel-injection.spel-injection

TABLE X: Details of token consumption, analysis runtime for all C/C++ projects

CWE Type	Repository Name	Size (KLOC)	Tool	Input Tokens (K)	Output Tokens (K)	Time (min)
CWE-401	linux/sound	1,253.23	RepoAudit	23748.25	5700.42	814.78
			KNighter	61.41	16.48	131.33
			CodeQL	-	-	4.08
			Semgrep	-	-	0.18
	linux/mm	133.95	RepoAudit	20009.25	4636.56	2450.41
			KNighter	61.41	16.48	128.53
			CodeQL	-	-	0.58
			Semgrep	-	-	0.07
	ImageMagic	680.88	RepoAudit	68233.39	3794.81	197.53
			KNighter	61.41	16.48	37.55
			CodeQL	-	-	3.39
			Semgrep	-	-	0.14
CWE-416	linux/net	962.14	RepoAudit	2735.89	102.44	7.02
			KNighter	185.25	47.92	219.84
			CodeQL	-	-	2.85
			Semgrep	-	-	0.80
	linux/drivers/net	3,924.62	RepoAudit	748.51	198.80	21.69
			KNighter	185.25	47.92	221.51
			CodeQL	-	-	14.08
			Semgrep	-	-	2.43
	vim	1,192.93	RepoAudit	24656.59	2916.95	67.18
			KNighter	185.25	47.92	123.62
			CodeQL	-	-	1.75
			Semgrep	-	-	0.45
CWE-476	linux/drivers/peci	1.72	RepoAudit	1547.98	211.58	30.65
			KNighter	127.52	45.26	196.73
			CodeQL	-	-	0.21
			Semgrep	-	-	0.07
	gpac	859.53	RepoAudit	38530.07	3413.36	132.64
			KNighter	127.52	45.26	132.20
			CodeQL	-	-	378.95
			Semgrep	-	-	0.20
	bitlbee	33.65	RepoAudit	225493.97	38,078,26	312.87
			KNighter	127.52	45.26	105.46
			CodeQL	-	-	0.37
			Semgrep	-	-	0.10

TABLE XI: Details of token consumption, analysis runtime for all Java projects

CWE Type	Repository Name	Size (KLOC)	Tool	Input Tokens (K)	Output Tokens (K)	Time (min)
CWE-022	OpenOLAT	1,907.08	IRIS	1,752.77	267.13	103.40
			CodeQL	-	-	15.69
			Semgrep	-	-	1.96
	spark	11.94	IRIS	8.47	1.45	2.55
			CodeQL	-	-	3.50
			Semgrep	-	-	0.61
	Dspace	448.03	IRIS	676.20	95.93	37.03
			CodeQL	-	-	15.42
			Semgrep	-	-	0.94
CWE-078	xstream	76.34	IRIS	116.92	211.99	10.70
			LLMDFA	18,716.58	778.23	993.00
			CodeQL	-	-	1.74
			Semgrep	-	-	0.58
	workflow-cps-plugin	238.59	IRIS	91.55	20.06	8.02
			LLMDFA	2,748.90	69.50	44.00
			CodeQL	-	-	3.27
			Semgrep	-	-	0.75
	tika	228.31	IRIS	468.01	99.10	32.45
			LLMDFA	18,499.28	1,459.38	1654.00
			CodeQL	-	-	9.24
			Semgrep	-	-	0.90
CWE-079	xwiki-platform	5,553.10	IRIS	927.21	188.67	60.48
			LLMDFA	23,864.02	1,994.81	4638.00
			CodeQL	-	-	12.33
			Semgrep	-	-	1.12
	jenkins	300.49	IRIS	463.74	79.21	24.32
			LLMDFA	2,535.12	245.51	530.00
			CodeQL	-	-	4.64
			Semgrep	-	-	0.38
	keycloak	6,033.86	IRIS	921.60	154.35	177.87
			LLMDFA	38,310.33	2,646.89	4141.00
			CodeQL	-	-	14.33
			Semgrep	-	-	0.57
CWE-094	onedev	552.49	IRIS	172.56	36.25	11.32
			CodeQL	-	-	7.15
			Semgrep	-	-	0.61
	activemq	611.96	IRIS	488.44	85.97	2089.47
			CodeQL	-	-	6.78
			Semgrep	-	-	0.39
	cron-utils	16.24	IRIS	17.79	3.08	3.67
			CodeQL	-	-	0.96
			Semgrep	-	-	0.24
CWE-722	sql2o	8.03	INFERROI	142.50	64.51	20.25
			CodeQL	-	-	1.49
	RxJava	325.39	INFERROI	2,871.49	1,408.23	176.27
			CodeQL	-	-	7.23
	jsoup	38.13	INFERROI	566.37	254.78	37.76
			CodeQL	-	-	1.04