Tool name	Team member who did the analysis	Commercial or Open Source	Test Code used	Did it run?	Did it cover all the code it was tested on (as far as you could tell)? Yes, No, Not sure	If not, why (briefly)?	Did it find all known buffer overflow vulnerabilities in the test code? Yes, No, Not sure	If not, why (briefly)?	Did it find all SQL/command injection vulnerabilities in the test code? Yes, No, Not sure	If not, why (briefly)?	Link to full analysis document
SonarQube	Chinyere	Both	~/soecm_data/REAL_DATASET	yes	yes		Yes		Not Sure	Flags SQL Injection, however unsure if all found	Link
CodeQL	Jeff	Commercial	CPP: https://github.com/Jeffrey-Matthew/testCodeQL Python: https://github.com/Jeffrey-Matthew/testpyCodeQL	Yes	Yes		No	For python, it is not able to cover because it is not able to run the buffer overflow query	No	For python, it is not able to cover because it is not able to run the SQL Injection query	Link
CodeScene	Jason	Commercial	C/C++ Juliet 1.3 test suite: https://github.com/jgarciasol/C gathered from https://samate.nist.gov/SARD/test-suites/112 **~/soecm_data/juliet_c_cpp**	Yes	Yes	-	No	Not designed to do so. CodeScene is a behavioral code analysis tool. It's main focus is on development time and maintenance cost.	No	Not designed to do so.	Link
Coverity Scan	Jason	Commercial	C/C++ Juliet 1.3 test suite: https://github.com/jgarciasol/C gathered from https://samate.nist.gov/SARD/test-suites/112 **~/soecm_data/juliet_c_cpp/testcases/CWE121_Stack_Based_Buffer_Overflow/s01**	Yes	Yes	-	No	Was not able to pick up some of the defects. Detected other defects that were not BoF.	No	Out of 471 units that contained OS command injections, it detected 154 over defects. 144 being OS command injections	Link
CPPCheck	Drew	Open Source	**`soecm_data/REAL_DATASET`**	Yes	Yes	-	No	The tool attempts to find the least amount of false positives as possible, so it will not report things it is not highly confident in.	No	Not designed to do so.	Link

Bandit	Drew	Open Source	**`soecm_data/vulnerable_python_project,`**
			`soecm_data/bandit/examples`
				Yes	Yes	-	No	Not designed to do so.	Not sure	Bandit flags many SQL injections, but the description of the vulnerable python project does not detail how many there are.	Link

DevSkim	Drew	Open Source	**`soecm_data/DATA_DIR,`**
			`soecm_data/REAL_DATASET`
				Yes	Yes	-	Not sure	Found virtually all poor uses of functions that would cause buffer overflow except for **`sprintf()`**. However, DevSkim will sometimes report it has found something without stating the name. It is possible this could be **`sprintf()`**.	No	Unable to detect these functions with provided basic rules.	Link

CodeHawk-C	Jeff	Open Source	/soecm_data/CodeHawk_test/CodeHawk-C/TEST_DIR/TEST_DIR	Yes	No	It doesn't work on project module.	Yes		No	Not designed to work on commands which cause SQL Injection.	Link
PMD	Jason	Open Source	**~/soecm_data/DATA_DIR/sql-inj.java,** **BenchmarkJava: https://github.com/OWASP-Benchmark/BenchmarkJava** **~/soecm_data/Benchmark**	Yes	Yes	-	No	Not designed to do so	No	PMD has no security rulesets. It is only able to detect programming mistakes	Link
SonarLint	Chinyere	Open Source	~/soecm_data/REAL_DATASET	yes	Not Sure	No Results, however syas time it took to process all files	No	No Results	No	No Results	Link