This work package presents an information exposure checker which is designed to detect information exposures in C/C++ code. Information flow vulnerabilities in C code are detrimental as they can cause data leakages or unexpected program behavior. Detecting such vulnerabilities with static code analysis techniques is challenging because of complex control and data flow. Static analysis tools used for detecting information exposure bugs can help software engineers detecting bugs without introducing run-time overhead. Such tools can make the detection of information-flow bugs faster and cheaper without having to provide user input in order to trigger the bug detection. We present a bug-detection tool for detecting information exposure bugs in C/C++ programs. Our tool is context-sensitive and uses static code analysis for bug detection, which was developed in the SIBASE working package 5.2.1. We developed our bug finding tool as an Eclipse plugin in order to easily integrate it in software development work flows. Textual annotations introduce information flow constraints into code as described in the SIBASE working package 5.1.2. The constraints are checked later by our tool. The bug reports provide user friendly visualizations that can be easily traced back to the location where the bug was detected. We discuss one static analysis approach for detecting information exposure bugs and relate briefly the usability of our bug testing tool to empirical research. We conducted an empirical evaluation based on 90 test programs which were selected from the National Institute of Standards and Technology (NIST) Juliet test suite for C/C++ code. We reached a true-positive coverage of 94.6% in ≈121 seconds for the test programs. Our results show that our approach is effective and can be further applied to detection of different types of vulnerabilities. This report is based on publications listed in Chapter 6.