Dynamic code analysis

Business

The objectives of the dynamic code analysis are to minimize the debugging time and to automatically pinpoint towards the potential errors and explain them as they occur during the program’s execution The aim of this paper is to emphasize the benefits of a dynamic code analyzer realized by implementing a virtual processor that simulates the code execution, together with the detection of the potential execution errors: out of memory requests, infinite loops, recursive calls etc.

Moto “ How hard would an auto mechanic’s Job be if they were never allowed to turn on the car and listen to hear what kind of weird sound it was making, or plug in the diagnostics check while the car was running. That’s dynamic analysis. ” [Eric JARVI] l. Introduction While developing software applications, programmers have to make their rograms reliable, and this considering a large number of scenarios and possible confgurations.

The dynamic code analysis can discover these types of errors and directly pinpoint to the vulnerable spots in the code.

The dynamic code analysis represents the investigation of a program’s behavior using the information obtained while the program is executing. The main goal of the dynamic code analysis is to establish which sections of the programs contain potential errors that are not usually found by compiler or by means of static analysis. and to automatically pinpoint to the potential errors and explain them as they occur. The dynamic code analysis proves itself to be useful, especially because many of the errors within a program are not found during regular testing.

.

Within a code there can be found very subtle errors, such as memory corrupted areas that can run perfectly on a software platform, but be a disaster on others. The large variety of the errors concerning the memory access can cause serious problems during production, especially of we consider that in embedded systems the security and stability are The result can be materialized in an event log or in a log of the statistically observed events. Dynamic analysis tools are all about helping humans understand the system in the holistic sense, while it is running" [Eric JARVI] II.

Static analysis vs. dynamic analysis The static code analysis is the analysis of a software that is performed without executing the program.

In most cases the analysis is performed on a certain version of the source code, but it can sometimes be performed on the object code. The term " static analysis" is usually attributed to some automatic program that executes the analysis; the analysis performed by the programmer is rather called " understanding the program".

The term " static" is due to the fact that the program is not actually executed, but rather tested in an analytical manner. The approaches based in the static analysis operate with static representations of the program and considers all the possible behaviors (out of which many will be impossible to achieve) In the opposite corner, the dynamic approaches analyze only the observed behaviors. These include " offline" techniques that operate with representations of the system's behavior, but also include a set of execution techniques that analyze the system's behavior " on the fly'.