

Project Description: Online TA

Master Project

Datalogisk institut, Copenhagen University (DIKU)

Oleksandr Shturmov

oleks@diku.dk

February 4, 2014.

We would like to specify and implement an online system for automatically evaluating submissions to assignments. The prime focus is on programming assignments, the submissions to which tend to lend their way to automated evaluation.

The system is meant as an aid to the traditional on-site education at the Department of Computer Science, University of Copenhagen. The intent of the system is to aid teachers and their assistants in some of the evaluations they have to make in the course of a programming course. It would be fair to say that we would like to design and implement an online teacher's assistant, or online TA.

The fact that the producers and the consumers of the learning content will be Computer Science teachers and students, allows us to disregard many of the problems that would otherwise be worth dealing with, e.g. graphical user interfaces. In contrast, the evaluation of submissions to programming assignments opens the system up to challenging problems and threats. The automatic evaluation of such submissions typically involves running static or dynamic analyses of arbitrary student programs.

We would like to ensure that all evaluations run in isolated environments, and all students get a fair share of system resources. Student programs may run for too long, use too many resources, fail in unpredictable ways, or even deliberately attempt to compromise the system. The analyses themselves, specified by the teachers and their assistants, may behave likewise, except the lack of incentive to deliberately compromise the system.

It is these problems and threats which are the focus of this project. In the following sections we further specify the scope of the project, and present the project-specific learning objectives.

Keywords: online, automatic evaluation, security engineering, arbitrary program execution, learning management system.

Scope

The implemented system should be capable of handling an on-site programming course at DIKU. This demands a moderate-sized web-based platform, capable of handling a couple hundred simultaneous submissions.

As we are dealing with Computer Science teachers and students, a command-line interface with the web-application would suffice. The design however, should support the latter addition of a graphical user interface, especially for students.

The programs submitted by students and the analyses specified by teachers and their assistants may, for all intents and purposes, be regarded as arbitrary programs that may or may not

adhere to some posed requirements. It may however be assumed that the teachers and assistants have no incentive to deliberately compromise the system. Furthermore, we would like to stay programming language agnostic, so we constrain ourselves to the operating system level.

The system should enforce read and write permissions, disallowing students to pose or change assignments, submit on other students behalf, view or change the private information of teachers, their assistants, or other students, etc.

The system should stay programming language agnostic and allow the use of various existing tools for the static and dynamic analyses of student programs, e.g. compilers or linters. The use of tools installed natively on the system poses its own security risks.

These are some the basic building blocks of a learning management system [ASTD, LMS].

Learning objectives

The following is a list of project-specific learning objectives:

1. Perform a security analysis, and identify safe security policies for
 - (a) an online system;
 - (b) arbitrary program execution; and
 - (c) a basic learning management system.
2. List, explain, and compare possible measures for enforcing the security policies above.
3. Design and implement a system which enforces the security policies above.
4. Verify that the system mediates the threats identified in the analysis above, using e.g.:
 - (a) unit-testing;
 - (b) property-based testing;
 - (c) stress testing (e.g. many simultaneous submissions); and
 - (d) penetration testing.

References

- [ASTD, LMS] Ryan K. Ellis. *A Field Guide to Learning Management Systems*. 2009. Learning Circuits. American Society for Trading & Development. Retrieved from http://www.astd.org/NR/rdonlyres/12ECDB99-3B91-403E-9B15-7E597444645D/23395/LMS_fieldguide_20091.pdf on February 1, 2014.