Project Proposal: Detection and Mitigation of Dependencies with Security Risks

Gleb Naumenko
University of British Columbia
Vancouver, Canada
naumenko@cs.ubc.ca

Puneet Mehrotra
University of British Columbia
Vancouver, Canada
puneet89@cs.ubc.ca

Anna Scholtz
University of British Columbia
Vancouver, Canada
ascholtz@cs.ubc.ca

ABSTRACT

Many code bases rely on outdated dependencies that might pose security risks. We propose to develop a tool that can automatically detect dependencies with security risks, runs a build with updated dependencies, and informs the developer about potential upgrade errors. We plan to use the tool to analyze a range of Github projects in order to answer several research questions. We also aim to do a user study on the usability and usefulness of the proposed solution.

1 INTRODUCTION

Code often relies on outdated dependencies or uses deprecated methods that might pose critical security issues. As is often the case, these projects might have only a limited number of resources working on them – for example in a low priority legacy product – or might have no active developers, but a strong user-base, as is the case in many open source projects. In these situations it is important to update these dependencies and rewrite insecure code, however doing so can introduce bugs in the application. Resolving these errors requires some development effort.

It can be useful to have a tool that automatically identifies these dependencies and potential upgrade errors, thereby assisting in proactively making decisions about the upgrade plan.

2 GOALS

The goal of our project is to build a tool that:

- Detects dependencies which have issues with security or performance
- Identifies locations where unsafe methods with known vulnerabilities are used in the code and suggests safe alternatives
- Helps developers to update dependencies in their projects. This includes propositions of updates and checking if updating breaks the code

As a prototype, the tool will be able to detect risks in Python projects but can be designed to support more programming languages in the future. We plan to implement it as a command-line tool as well as an IDE plugin.

For detection the tool will rely on a database that contains entries for known problematic dependencies and a description of their associated risks. The tool will extract all dependencies from the provided source code, match them against the database and generate a report for the developer. In addition, it also offers to update the dependencies if fixed versions are available. The update will run on a clone of the source code and also generate a report which will contain whether the update was successful or provide information about what kind of errors occurred.

3 RESEARCH QUESTIONS

Using the developed tool we want to analyze different projects on Github and address the following research questions:

- How many popular projects have dependencies that pose security risks?
- What are the most common risks?
- How easily can these projects be updated to eliminate the risks?

4 EVALUATION

To evaluate the tool we plan to select trending Github repositories, analyze them and answer the research questions. We also plan to run a small user study in which we ask participants to use the tool and participate in an interview. The interview will focus on the usefulness and usability of the developed tool.