SCIENCES & BIOENGINEERING SCIENCES

The Research Group
Software Languages Lab

has the honor to invite you to the public defence of the PhD thesis of

Ruben Opdebeeck

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:

Static Analysis for Quality Assurance of Ansible Infrastructure-as-Code Artefacts

Promotor:
Prof. dr. Coen De Roover

The defence will take place on

Friday October 25, 2024 at 17:00 p.m.

VUB Etterbeek campus, Pleinlaan 2, Elsene in auditorium I.2.01.

Members of the jury

Prof. dr. Viviane Jonckers (VUB, chair)
Prof. dr. Antonio Paolillo (VUB, secretary)
Prof. dr. Kris Steenhaut (VUB)
Prof. dr. Beat Signer (VUB)
Prof. dr. Dimitris Mitropoulos (National and Kapodistrian University of Athens, GR)
Prof. dr. João F. Ferreira (Universidade de Lisboa, PT)

Curriculum vitae

Ruben Opdebeeck obtained his MSc in Computer Science at the VUB in 2019. He then started a PhD at the Software Languages Lab (SOFT), supported by an SB fellowship from the Research Foundation – Flanders.
Ruben's research resulted in eight publications in international peer-reviewed journals and conferences (7 as first author), and one co-authored book chapter. He also presented his research at two industrial exhibitions and the "Dagstuhl" scientific seminar. In 2023, he spent 3 months at the MCIS lab of Queen's University, Canada where he conducted research under the guidance of Prof. Dr. Bram Adams. Next to his research, he also guided 10 bachelor thesis students and 8 master thesis students.

Abstract of the PhD research

Today's software-intensive industry uses increasingly complex digital computing infrastructures, possibly comprising hundreds of cloud resources such as virtual computing instances and managed databases. Manually managing such infrastructures is laborious and error-prone. Therefore, Infrastructure as Code (IaC) enables practitioners to automate the provisioning, configuration, and orchestration of infrastructures through executable code. The correctness and security of such code are vital, as defects can cause disastrous outages while security weaknesses leave the infrastructure vulnerable to cyberattacks. However, existing code quality assurance analyses for IaC are lacklustre and either underperform by not reasoning about the code's behaviour or are resource-intensive to apply. Moreover, they focus solely on the infrastructure code and ignore its supporting software supply chains, which form a major cyberattack vector today.

This thesis aims to alleviate these issues through four contributions aimed at Ansible, one of the most widely-used IaC tools. The first is the Program Dependence Graph (PDG) for Ansible, a graph representation of infrastructure code that captures its behaviour. We introduce an analysis to extract a PDG for Ansible scripts statically, i.e., without executing the code, thereby bridging the gap between lightweight yet underperforming approaches and accurate yet resource-intensive approaches.

The second contribution is an application of the PDG representation to detect behavioural "code smells" that may be indicative of defects in infrastructure code. By detecting patterns in these graphs, the analysis can uncover six error-prone coding practices related to Ansible variables.
We apply this analysis in a large-scale empirical study to investigate the prevalence and lifetime of these smells in open-source Ansible code.

GASEL, the third contribution, is an application of the PDG representation to detect security weaknesses in IaC. It uses graph queries on the PDG to identify seven types of "security smells" that may lead to security weaknesses or attack vectors. The behavioural information encoded in the graphs enables GASEL to outperform state-of-the-art security smell detectors. We apply GASEL in a large-scale empirical study investigating security smells in open-source Ansible code.

The fourth contribution is an empirical study of Ansible's deployment software supply chain. We propose an automated analysis that identifies third-party dependencies of Ansible plugins, and we apply it to study the types of dependencies that occur in practice.

The empirical studies presented in this thesis call for improvements to quality assurance practices in Infrastructure as Code, while the proposed static analyses pave the way for advanced tooling to this end.