**INTRODUCTION**

In many companies, private clouds are considered to be an important element of data center transformations. Private clouds are dedicated cloud environments created for the internal use by a single organization . According to the Cloud Survey 2017 ,private clouds are adopted by 72% of the cloud users, while the hybrid cloud adoption (both public and private) accounts for 67%. The companies, adopting private clouds, vary in size from 500 to more than 2000 employees. Therefore, designing and developing secure private cloud environments for such a large number of users constitutes a major engineering challenge. Usually, cloud computing services offer REST APIs (REpresentational State Transfer Application Programming Interface) to their consumers. REST APIs, e.g., AWS, Windows Azure , OpenStack , deﬁne software interfaces allowing for the use of their resources in various ways. The REST architectural style exposes each piece of information with a URI, which results in a large number of URIs that can access the system. Data breach and loss of critical data are among the top cloud security threats . The large number of URIs further complicates the task of the security experts, who should ensure that each URI, providing access to their system, is safeguarded to avoid data breaches or privilege escalation attacks. Since the source code of the Open Source clouds is often developed in a collaborative manner, it is a subject of frequent updates. The updates might introduce or remove a variety of features and hence, violate the security properties of the previous releases. It makes it rather unfeasible to manually check correctness of the APIs access control implementation and calls for enhanced monitoring mechanisms. In this paper, we present a cloud monitoring framework that supports a semi-automated approach to monitoring a private cloud implementation with respect to its conformance to the functional requirements and API access control policy. Our work uses UML (Uniﬁed Modeling Language) models with OCL (Object Constraint Language) to specify the behavioral interface with security constraints for the cloud implementation. The behavioral interface of the REST API provides an information regarding the methods that can be invoked on it and pre- and post-conditions of the methods. In the current practice, the pre- and post-conditions are usually given as the textual descriptions associated with the API methods. In our work, we rely on the Design by Contract (DbC) framework , which allows us to deﬁne security and functional requirements as veriﬁable contracts. Our methodology enables creating a (stateful) wrapper that emulates the usage scenarios and deﬁnes security-enriched behavioural contracts to monitor cloud. Moreover, the proposed approach also facilitates the requirements traceability by ensuring the propagation of the security speciﬁcations into the code. This also allows the security experts to observe the coverage of the security requirements during the testing phase. The approach is implemented as a semi-automatic code generation tool in Django – a Python web framework – and validated using OpenStack as a case study. OpenStack is an open source cloud computing framework providing IaaS (Infrastructure as a Service) . The validation using OpenStack has shown promising results and motivates us to continue the tool development described in this paper. The paper is organized as follows: section II motivates our work. Section III gives an overview of our cloud monitoring framework. In section IV, we present our design approach to modelling stateful REST services. The contract generation mechanism is described in section V. Section VI presents the tool architecture and our work with monitoring OpenStack. The related work and the conclusion are presented in sections VII and VIII correspondingly.