**Generating Cloud Monitors from Models to Secure Clouds**

Authorization is an important security concern in cloud computing environments. It aims at regulating an access of the users to system resources. A large number of resources associated with REST APIs typical in cloud makes an implementation of security requirements challenging and error-prone. To alleviate this problem, in this paper we propose an implementation of security cloud monitor. We rely on model-driven approach to represent the functional and security requirements. Models are then used to generate cloud monitors. The cloud monitors contain contracts used to automatically verify the implementation. We use Django web framework to implement cloud monitor and OpenStack to validate our implementation.

**EXISTING SYSTEM:**

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. 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. 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.

**DISADVANTAGES OF EXISTING 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.

**PROPOSED SYSTEM:**

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.

**ADVANTAGES OF PROPOSED SYSTEM:**

* Our methodology enables creating a (stateful) wrapper that emulates the usage scenarios and defines security-enriched behavioural contracts to monitor cloud.
* 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.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 500 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 1GB.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows 7.
* Coding Language : Python
* Tool : PyCharm, Visual Studio Code
* Database : MYSQL

**REFERENCE:**

Irum Rauf A bo Akademi University,Turku, Finland, Elena Troubitsyna KTH – Royal Institute of Technology, Stockholm, Sweden, “**Generating Cloud Monitors from Models to Secure Clouds**”, Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) IEEE Xplore: 23 July 2018, DOI: 10.1109/DSN.2018.00060