Creating Cloud Monitors from Models, Securing Clouds

1Prof.Mayank Mangal, 2Shubham Singh, 3Suraj Tawale, 4Akhil Singh

1Head of Department, Department of Computer Engineering, Alamuri Ratnamala Institue of Engineering and Technology
2,3,4Student, Department of Computer Engineering, Alamuri Ratnamala Institue of Engineering and Technology

---***---

ABSTRACT

In cloud computing, permission is a critical security risk. Its goal is to limit user access to the network. The large number of resources associated with REST APIs found in the cloud makes implementing security requirements difficult and error-prone. Security cloud monitors are employed to solve the issue. The performance and security needs are represented using a model-driven methodology. Cloud monitors are created using models. Contracts are included in the cloud monitors, which automatically verify the execution. The cloud monitor is implemented using the Django framework, and the solution is validated using OpenStack

KEY WORDS: Cloud Computing, REST APIs, Cloud Monitoring Framework.

INTRODUCTION

Consumers of cloud computing services have access to REST APIs. REST APIs, like as those provided by AWS, Windows Azure, and OpenStack, establish software interfaces that allow users to access their resources in a variety of ways. Each piece of information in the REST architectural style is exposed using a URI[8][9], resulting in a vast number of URIs that can access the system. The top cloud security dangers are data breaches and the loss of crucial data. The vast quantity of URIs complicates the duty of security specialists, who must guarantee that each URI that grants access to their system is protected to prevent data breaches or privilege escalation attacks. Because the Open Source clouds' source code is frequently updated in a collaborative fashion, it is prone to frequent modifications. The modifications may add or delete a range of features, and thereby compromising the prior release' security qualities. It makes it difficult to manually check the API access control implementation's accuracy, necessitating the use of improved monitoring techniques.

It specifies the behavioural interfaces with security policy for the implementations using UML (Unified Modelling Language) models and OCL (Object Constraint Language). The REST API's behavioural interface[8] provides information about the methods that can be called on it, as well as the methods' pre- and post-conditions. The pre- and post-conditions are commonly specified in the API methods' descriptions. It is based on the DbC framework[10], which allows security and functional requirements to be defined as verified contracts.. The methodology enables creating a (stateful) interface for cloud monitoring that simulates application scenarios and defines security-enhanced behavioural contracts.

AIM

The goal of this work is to create a cloud monitor that can authorise information saved in the cloud utilising REST methods. The creation of a control monitor is dependent on the design model produced by the application owners. Its goal is to regulate nursing assistants' access to system resources. Because of the large number of resources associated to REST APIs that are common in the cloud, implementing security requirements is complex and fallible.

OBJECTIVE

The paper's aim is to keep track of users' (employees') actions on the system. Also, traffic on the system's networking is monitored, and the system is protected from cyber-attacks. Using cloud monitoring to improve the security of a cloud system.

Log-based cloud monitoring system for OpenStack:

Cloud computing is a rapidly growing remote computing framework that provides customers with on-demand computational resources and services over the Internet. OpenStack is an example of an open-source platform that aids in the creation of private clouds for businesses. Tracking cloud operations is a critical responsibility, and log files are frequently used to do so. Log management is critical for diagnosing and troubleshooting various issues in the cloud network.. Managing logs can be time consuming and difficult to read. It is proposed in the paper that a system monitors the logs of OpenStack components (i.e., Nova, Neutron, Cinder) in real-time and generates an alert for information, debug, error, warning, and trace messages as soon as an issue is detected in the logs. It makes it easier for administrators to read. Apart from that, the logic aids in the generation of warning messages prior to the various resource quotas, such as virtual CPUs, Floating IPs, and so on. A tool like this can assist the administrator in increasing or releasing quotas, hence reducing the nuisance to users who build virtual machines. It constructs the suggested system and uses simulation to demonstrate its feasibility and performance.

Industrial Cloud Monitoring System Based on Internet of Things:

Cloud computing and Internet of Things skills can help firms monitor their assets. The study provides a new method for monitoring machines that is based on a low coast board (Omega2 Plus). The Omega2 Plus is an integrated device that functions as a computer and has the shape and size of a credit card. It has the ability to send data directly to the cloud server. Multiple types of machines exist in any industrial factory; the solution attached machines' logical outcomes to Omega2 Plus GPIO and sent data to a web-based server to be monetarized. The Cloud Monitoring System (CMS) provides real-time statistics on production data[3], current work, employee IDs working on specific machines, production rate, and downtime. Supervisors or professionals in nearly any region of the world keep an eye on all of the machines. Furthermore, the Cloud Monitoring System is simple and may be operated by an inexperienced worker on a basic level.

EXISTING SYSTEM

Private clouds are seen as a key part of data centre conversions in many enterprises. Private clouds are private cloud environments developed for a single organization's internal use. As a result, creating secure and private cloud environments for such a huge number of users is a significant engineering issue. REST APIs (Representational State Transfer Application Programming Interface) are typically provided by cloud computing services to their customers. Each item of information is exposed using a URI in the REST architectural style, resulting in a vast number of URIs that access the service.

PROPOSED SYSTEM

The cloud monitoring framework that enables a semi-automated approach to tracking a private cloud implementation's compliance with requirement specification and API access control policies. The work specifies the behavioural interface with security constraints for the cloud implementation using UML (Unified Modeling Language) models and OCL (Object Constraint Language). The REST API's behavioural interface displays the information about the methods that can be called on it, as well as their pre- and post-conditions. The pre- and post-conditions are commonly specified in the API methods' textual descriptions. It is based on the Design by Contract (DBC) framework, which allows security and functionality needs to be defined as verified contracts.. To monitoring the cloud, the methodology allows for the creation of a (stateful) framework that emulates usage scenarios and defines security-enriched behavioural contracts. The suggested approach also makes requirements tracing easier by guaranteeing that security standards are propagated into the code. During the testing process, it also allows security specialists to monitor the coverage of the security standards. The method is implemented as a tool for semi-automated code generation.

THE PUSH-PULL ALGORITHM

Step 1: While(true)

Step 2: Set pull operation identifier is pulled<- false

Step 3: Waiting for the termination of the push_interval

Step 4: If is Pulled equals to true during Push_interval

Step 5: Push current update to master node

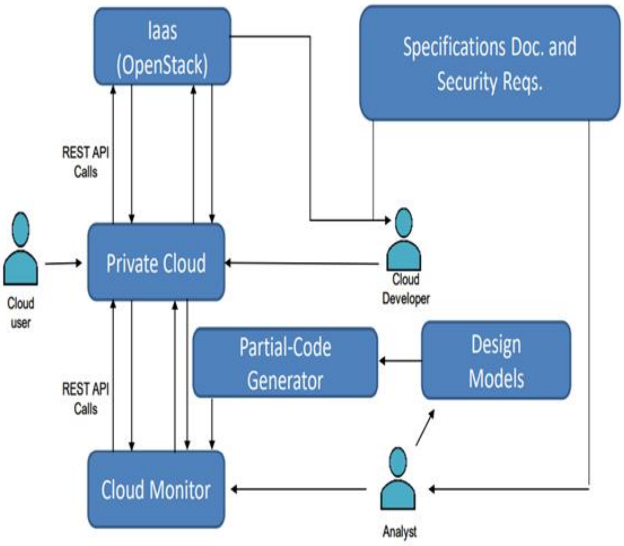
Step 6: else if (| current_value – previous_value | / (MAX - MIN)≥ UTD)

Step 7: IsPushed<- false

Step 8: Push current update to master node

Step 9: //end while

SYSTEM ARCHITECTURE



The Cloud Monitoring Framework's overall design in IaaS is utilised by a cloud developer to create a private cloud for her or his company that will be used by various cloud users within the company. A group of developers working cooperatively on various workstations may be able to implement the private cloud in some circumstances. The IaaS REST API is used to build the private cloud in accordance with the specification document and security policy. On top of the private cloud, the cloud monitor is installed.

The primary original components of the work are highlighted in the boxes below. Based on the specification document and security policies, the security analyst creates the necessary design models. These models are used to define the functional and security criteria for the private cloud's behavioural interface. Furthermore, utilising REST as the underlying stateless architecture, the design models describe all of the information needed to develop stateful scenarios. Each item of information is exposed using a URI in the REST architectural style, resulting in a vast number of URIs that can interact with the system.

ADVANTAGES

1) To create APIs with REST interface features, it uses a model-driven method.

2) The proposed semi-automated technique relies on modelling rather than human code inspection or testing to assist cloud developers and security specialists in identifying security loopholes in the implementation.

3) It aids in the detection of faults that could be used in data breaches or privilege escalation attacks.

4) Openstack, an open-source cloud computing platform, offers private cloud services for free.

5) Both users and administrators will find the system to be very user-friendly.

CONCLUSION

We have proposed an approach and related tool for cloud security monitoring in this work. In order to create APIs with REST interface features, we used a model-driven method. The models' cloud monitors provide for automatic contract-based verification of the correctness of performance and security requirements executed by a private cloud infrastructure. By focusing on models rather than manual code inspections or tests, the proposed semi-automated approach aimed to assist cloud developers and security specialists in identifying security gaps in the implementation.. It aids in the detection of faults that could be used in data breaches or privilege escalation attacks. Because open source cloud frameworks are frequently updated, our approach's automated nature allows developers to focus on modelling rather than manual code examination or testing. It aids in the detection of faults that could be used in data breaches or vulnerabilities attacks

REFERENCE

[1] Agrawal, Vaibhav, et al. "Log-based cloud monitoring system for OpenStack." 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, 2018.

[2] Al-Saedi, Ibtesam RK, and Saif Aldeen Saad Obayes Al-Kadhim. "Industrial Cloud Monitoring System Based on Internet of Things." 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). IEEE, 2018.

[3] OpenStack Block Storage Cinder. https://wiki.openstack.org/wiki/ Cinder. Accessed: 26.03.2018.

[4] Gao, Zefeng, and Xiaoyong Li. "A framework for monitoring and security authentication in cloud based on Eucalyptus." 2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). IEEE, 2015.