Privacy-Preserving Data Sharing and Adaptable Service Compositions in Mission-Critical Clouds

Bharat Bhargava[a], Pelin Angin[b] and Rohit Ranchal[c]

[a] *Purdue University, West Lafayette, IN, USA*
[b] *Middle East Technical University, Ankara, Turkey*
[c] *IBM Cloud Lab, Austin, TX, USA*

Abstract

Existing cloud systems lack robust mechanisms to monitor compliance of services with security and performance policies under changing contexts, and to ensure uninterrupted operation in case of failures. On the other hand, microservices-based cloud system architectures that have become indispensable for defense applications require systematic monitoring of service operations to satisfy their resiliency and antifragility goals. In this work we propose a unified model for enforcing security and performance requirements of mission-critical cloud systems even in the presence of anomalous behavior/attacks and failure of services. The model allows for proactive mitigation of threats and failures in cloud-based systems through active monitoring of the performance and behavior of services, promising achievement of resiliency and antifragility under various failures and attacks. It also provides secure dissemination of data between services to ensure end-to-end secure operation of critical missions.

Keywords 1

Cloud computing, privacy, service composition

1. Introduction

The rise of cloud computing and Internet of things (IoT) have created new security challenges with a large attack surface. Microservices-based cloud system architectures for defense applications require systematic monitoring of service operations to satisfy their resiliency (withstand cyber-attacks, and sustain and recover critical function) and antifragility (increase in capability, resilience, or robustness as a result of mistakes, faults, attacks, or failures) goals.

When clients interact with a cloud service, they expect certain levels of Quality of Service (QoS) guarantees, expressed as service performance, security and privacy policies. Controlling compliance with service level agreements (SLAs) requires continuous monitoring of services in an enterprise, as sudden changes in context can cause performance to deteriorate, if not result in the failure of a whole composition. To provide optimal performance in the enterprise cloud architecture under varying contexts, we need context-awareness and adaptation mechanisms for SOA and cloud service domains. Cloud platforms are vulnerable to increasingly complex attacks that could violate the privacy of data stored on them or shared with web services, which is especially detrimental in case of mission-critical operations. In order to mitigate these problems, cloud systems need to integrate proactive defense mechanisms, which provide increased resiliency by treating potentially malicious service interactions and data sharing before they take place.

These requirements call for the development of unified models for performance and security monitoring of operations that provide valuable input for achieving situation-awareness,

International Semantic Intelligence Conference (ISIC 2021), Feb 25-27, 2021, New Delhi, India
EMAIL: bbshail@purdue.edu (A. 1); pangin@ceng.metu.edu.tr (A. 2); ranchal@us.ibm.com (A. 3)

dynamic adaptability and restoration of services in the face of changes in context, and effective mechanisms for detection and sharing of threat data, as well as enforcing cross-domain security and Quality of Service (QoS) constraints. Controlled privacy and integrity-preserving data dissemination and filtering models are needed to ensure protection of the privacy of sensitive data in trusted and untrusted clouds.

In this paper, we describe the design of a unified monitoring and response model for privacy-preserving data dissemination and adaptable service compositions in mission-critical cloud systems. Through unsupervised learning-based detection of anomalies in cloud services and adaptable real-time service composition, the proposed model aims to achieve a highly resilient cloud architecture for mission-critical systems.

2. Related work

Current industry-standard cloud systems such as Amazon EC2 provide coarse-grain monitoring capabilities (e.g. CloudWatch) for various performance parameters for services deployed in the cloud. Although such monitors are useful for handling issues such as load distribution and elasticity, they do not provide information regarding potentially malicious activity in the domain. Log management and analysis tools such as Splunk [8], Graylog [9] and Kibana [10] provide capabilities to store, search and analyze big data gathered from various types of logs on enterprise systems, enabling organizations to detect security threats through examination by system administrators. Such tools mostly require human intelligence for detection of threats and need to be complemented with automated analysis and accurate threat detection capability to quickly respond to possibly malicious activity in the enterprise and provide increased resiliency by providing automation of response actions.

Development of runtime-auditing systems for mobile and web-based services has been the focus of many research efforts. Li et al. [3] describe a system for auditing runtime interaction behavior of web services. They use finite state automata to validate predefined interaction constraints, where message interception is bound to the particular server used for deploying Web services. Simmonds et al. [1] present a more comprehensive auditing solution for checking behavioral correctness of web service conversations. Their proposal is for a specific application server, since they utilize an event mechanism provided by that server.

To support flexible auditing of the behavior pattern for composite services, Wu et al. [2] demonstrate an “aspect extension” to WS-BPEL, in which history-based pointcuts specify the pattern of interest within a range, and advices describe the associated action to manage the process if the specified pattern occurs. Their solution addresses specific orchestration engines, which is not a generic solution for modern cloud-based services. In [3] and [4] the identification of trusted services and dynamic trust assessment in SOA are studied. Malik et al. [4] introduce a framework called RATEWeb for trust-based service selection and composition based on peer feedback. It is based on decentralized techniques for evaluating reputation-based trust with ratings from peers. Spanoudakis et al. [5] present an approach to keep track of trusted services to address the compliance of promises expressed within their service level agreements (SLAs). The trust assessment is based on information collected by monitoring services in different operational contexts and subjective assessments of trust provided by different clients. Approaches like [3] and [5] are not suitable for compositions with many services, as the monitoring system would need to collect intensive information from a lot of clients. Gamble et al. [6] present a tiered approach to auditing information in the cloud. Filtering and reasoning over the audit trails can manifest potential security vulnerabilities and performance attributes as desired by stakeholders. [7] introduces a system to model the essential security elements and define the proper message structure and content that each service in the composition must have, based on a security meta-language (SML). Both approaches focus on how services can comply with established standards, but their implementation requires extensive changes in the current infrastructures. Our previous work [17] proposed service interceptors to enforce policies on interactions between different cloud services in a composition. In this work, we take a monitoring approach for service health and anomalies for more informed real-time decisions and build on [16] to dynamically update service compositions with low overhead.

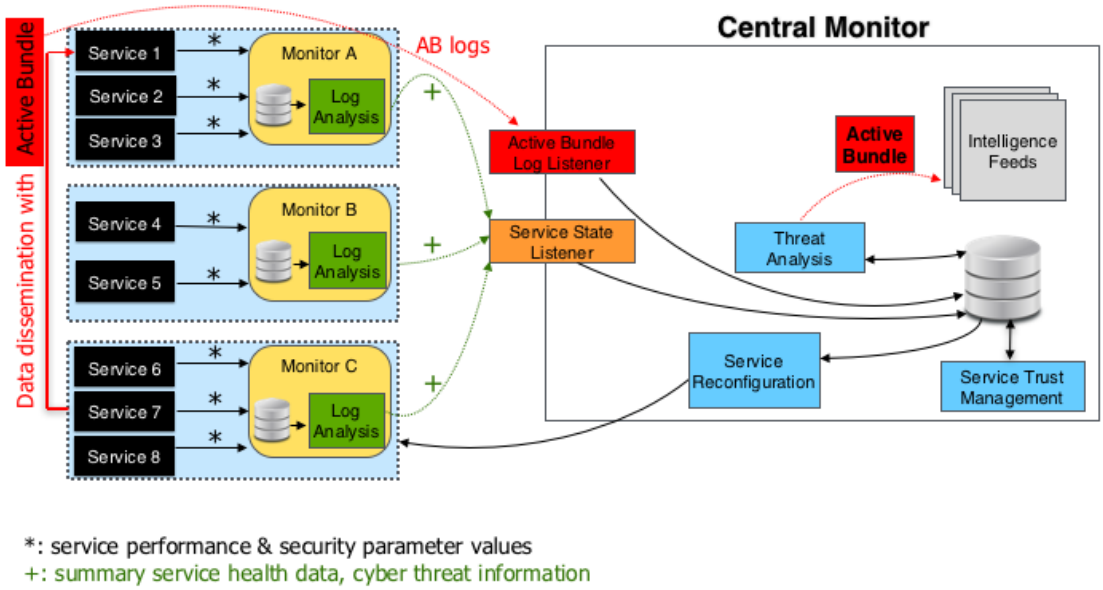

Figure 1: Solution architecture

3. Proposed Solution

In this paper, we describe an approach that uses a distributed network of *service activity monitors* to audit and detect service behavior and performance changes, adaptively update service compositions and securely share data in a mission-critical cloud system. By integrating components for service performance monitoring, dynamic service reconfiguration and adaptable data dissemination, the proposed model aims to provide a unified architecture for agile and resilient computing in trusted and untrusted clouds. The overall architecture of the proposed model is demonstrated in Figure 1. General characteristics of the solution are as follows:

- Each service domain, such as a cluster of machine instances in the cloud or a set of mobile services in close proximity to each other, has a service monitor that tracks interactions among the services in the domain as well as outside the domain.
- The local service monitors (Monitor A, Monitor B etc.) gather performance and security data including response time, response status, authentication failures, etc., among other parameters for each service by intercepting service requests and utilizing available performance monitoring software. The data collected are logged in the database of each local monitor and mined using unsupervised machine learning models to detect deviations from normal behavior. The analysis results are reported to a central monitor in the form of summary statistics for the services.
- The central monitor utilizes information submitted by local monitors to update trust values of services and reconfigure services/service compositions to provide resiliency against attacks and failures. The central monitor utilizes the gathered information to form cyber threat intelligence feeds about the services in a standard format.
- Detection of service failures and/or suboptimal service performance triggers restoration of optimal behavior through dynamic reconfiguration of service compositions.
- Privacy-preserving dissemination of data between services is achieved using active bundles. Likewise, data services in the cloud utilize active bundles for protected data storage that enforces fine-grain security policies associated with the usage of the data items when authorizing access.

3.1. Cloud Service Anomaly Detection

In this section we present our system architecture for the monitoring of cloud services and detection of anomalies in order to

provide adaptable and resilient service operation in a mission-critical cloud system. Figure 2 shows a high-level overview of service monitoring and anomaly detection in the proposed architecture.

Monitoring in the system architecture is distributed in the sense that each service domain, such as a cluster of machine instances in the cloud, has a service monitor that tracks interactions among the services in the domain as well as interactions with services or users outside the domain. When a service is deployed, it is registered with the local monitor of its domain in order to be discoverable by other services or users. The local monitors have access to all interactions with the services registered in their domain and they gather interaction/performance data streams containing items for response time, response status, authentication failures etc. among other parameters for each service using interceptors transparent to each service implementation. Services in each domain are also tracked using aspect-oriented programming (AOP)-based software monitors for parameters requiring finer-grained control. The data collected are mined by the anomaly detection module of the domain and reported to the central monitor in the form of summary health statistics and trust values for the services. These statistics are utilized by the dynamic service composition module when making decisions about which services to include in an orchestration.

3.1.1. Unsupervised learning for service anomaly detection

Research in machine learning has resulted in various models for detection of outliers in different types of data. While supervised and unsupervised classification models have been applied with success to a variety of domains [19], robust real-time models for detecting anomalies and failures in service operation are still in progress. The main shortcoming of supervised anomaly detection models including deep learning models is that they require a large amount of training data and can only provide accurate results on anomalies that were previously observed in the system. This makes such models unable to capture threats/anomalies that are completely new, which is essential in an environment of ever-growing security vulnerabilities and attacks.

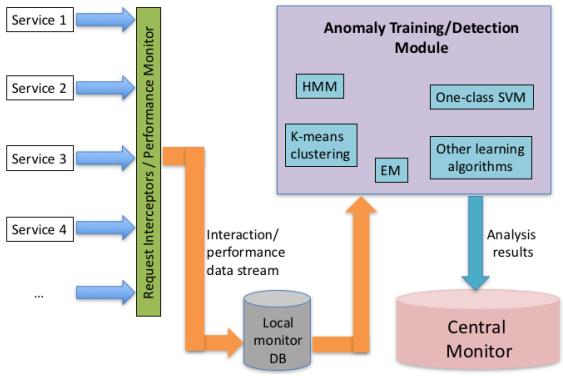

Figure 2: Cloud service anomaly detection

In this paper we focus on unsupervised models for outlier/anomaly detection in service behavior. A significant advantage of unsupervised models is that the training data required is gathered from the behavior of services operating under normal conditions (possibly in an isolated environment); i.e. no attack data is required to train these models. Specifically, we focus on two unsupervised learning models, k-means clustering and one-class support vector machines (SVM), due to their simplicity and success in anomaly detection tasks. Training of the models is performed with data gathered under normal system operation (i.e., isolated execution under a controlled runtime environment).

Service performance and security parameters that are used in the learning process for general cloud-based services and data services include: Number of requests/sec, total error rate, CPU utilization, memory utilization, number of authentication failures, number of connection failures, network latency, service response time, disk space usage, number of database connections. Note that this is not an exhaustive list and various other relevant parameters that can be obtained during service runtime through monitoring can be integrated into the learning algorithms easily.

K-means Clustering: K-means clustering partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean [11]. When applied to the service anomaly detection problem, k-means clustering finds clusters of parameter values of normal service behavior during the training phase, using the data obtained with service monitoring under normal operation. During the online anomaly detection process, data gathered by the service monitors are utilized to measure the distance of the service

behavior (i.e., values of performance/security parameters) at each time point to all clusters found by the algorithm. If the value does not fall in any cluster, an anomaly signal is raised.

One-class Support Vector Machines (SVM): One-class SVM [12] is an extension of the well-known support vector machines (SVM) classification algorithm, where training is performed using only positive examples and test instances are classified as belonging or not belonging to the single (positive) class. Essentially, one-class SVM learns a decision function for novelty detection, which is what we try to achieve in service anomaly detection to mitigate attacks with no well-known signature. SVM constructs a decision hyperplane boundary based on normal runtime conditions of the service it is trained for. During the online anomaly detection phase, instances lying outside the boundary for normal operation are classified as anomalous, resulting in an anomaly signal.

3.2. Privacy-Preserving Data Dissemination between Services in Mission-Critical Clouds

We propose a policy–based distributed data dissemination model, which provides secure data dissemination, i.e., every service gets access only to those parts of data for which it is authorized. The goal of the proposed solution is to selectively disclose information based on policies, minimize the unnecessary disclosure and ensure security and privacy of the information. Our solution uses Active Bundle (AB) to achieve this [13, 14, 15]. An active bundle (AB) is a self-protecting data mechanism that includes sensitive data, metadata (policies) and a policy enforcement engine (Virtual Machine) for policy enforcement. Clients interact with services by sending an AB, which contains encrypted data about their request and the policies associated with the data. AB is a data protection mechanism, which can be used to protect data at various stages throughout its lifecycle. AB is a robust and an extensible scheme that can be used for secure cross-domain data dissemination. AB includes the following components:

- **Sensitive data:** It is the digital content that needs to be protected from privacy violations, data leaks, unauthorized dissemination, etc. The digital content can include documents, pieces of code, images, audio, video files etc. This content can have several items, each with a different security/privacy level and an applicable policy to ascertain its distribution and usage.
- **Metadata:** It describes the active bundle and its policies. This can include information such as AB identifier, information about its creator and owner, creation time, lifecycle etc. It also includes policies that govern AB's interaction and usage of its data, such as access control policies, privacy policies, dissemination policies etc.
- **Policy Enforcement Engine (or Virtual Machine, VM):** It is a specific-purpose VM used to operate AB, protect its content and enforce policies (for example, disclosing to a service only the portion of sensitive data that it requires to provide service).

Further details of the active bundle solution can be found at [13].

4. Implementation of Distributed Service Monitoring and Adaptable Composition

In the prototype distributed service monitoring system, each local service monitor has been implemented using Apache Axis2 valves for intercepting all service requests in the domain and each service domain includes a MySQL database, in which data (response time, response status, CPU usage, memory usage) about each service gathered by the monitor is logged. Additionally, AOP-based service interceptors were added to allow for finer-grain monitoring and policy enforcement capability. The central monitor is implemented as a web service on Amazon EC2, which has its own database to store health, endpoint address and category data for various services. While each service invocation leads to an update in the local monitor's database, summary data for all services in a specific domain is reported to the central monitor periodically by each local monitor. One of the benefits of cloud computing is that there can be multiple options for services to achieve a specific task. We

define a *service category* as an abstraction for a set of services that provide similar functionality. A *service* is the actual implementation of the functionality for a specific service category. The dynamic service composition module utilizes information from the central monitor's database to create service orchestrations that comply with users' performance and/or security requirements on-the-fly. The goal of dynamic service composition is to maximize the resiliency and trustworthiness of the system based on selecting the best individual services, while meeting the constraints (security and SLA requirements).

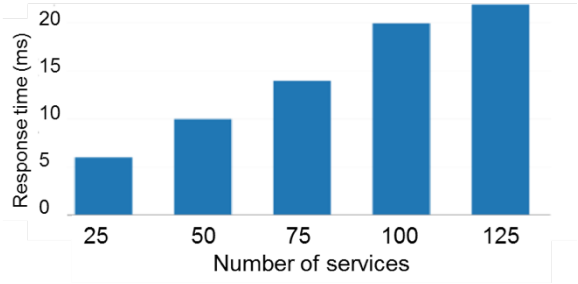

Figure 3: Effect of number of services on dynamic service composition time

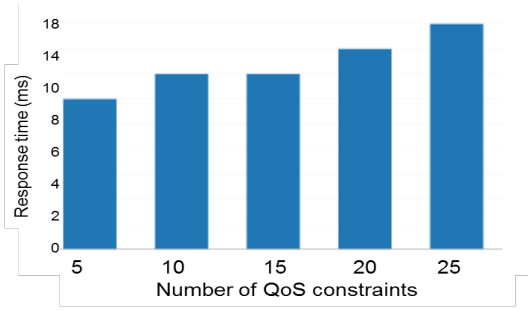

Figure 4: Effect of number of QoS constraints on dynamic service composition time

We performed experiments to evaluate the overhead of dynamic service composition using testbeds in the Amazon EC2 cloud. Note that the problem here is finding an optimal service composition (i.e., selecting a service from each service category required in the composition) subject to a set of QoS and security constraints. In the first experiment, we investigated the effect of the number of services to choose from for each service category, on the performance of dynamic service composition. In this experiment, we set the number of service categories to 5 and the number of QoS constraints to 3. Figure 3 shows the response time of the dynamic service composition module for scenarios with total number of services from 25 to 125. The results show that the execution time changes almost linearly. Even for 125 services in 5 categories (which is unlikely to be surpassed in any practical SOA scenario), the dynamic service composition module performs very well and the average response time is 22ms.

In the second experiment, we investigated the effect of the number of service constraints on the performance of dynamic service composition module. In this experiment, we set the number of services to 50 and the number of service categories to 5. According to Figure 4, the effect of the QoS constraints on performance is sublinear. Even after increasing the input size by a factor of 5, the response time only increases by 50% and remains under 20 ms.

5. Conclusion

Existing cloud enterprise systems lack robust mechanisms to monitor compliance of services with security and performance policies under changing contexts, and to ensure uninterrupted operation in case of failures. This work proposes a unified model for enforcing security and performance requirements of mission-critical cloud systems even in the presence of anomalous behavior/attacks and failure of services. Service monitors include components that enable the adaptation of the systems in response to detected anomalies, such that the non-stop system operations continue and comply with security requirements. The resiliency is accomplished through dynamic reconfiguration and restoration of services. Our approach is complementary to functionality provided by log management tools such as Splunk in that it develops models that accurately analyze the log data gathered by such tools to immediately detect deviations from normal behavior and quickly respond to such anomalous behavior in order to provide increased automation of threat detection as well as resiliency. Our approach allows for proactive mitigation of threats and failures in cloud-based systems through active monitoring of the performance and behavior of services, promising achievement of resiliency and antifragility under various failures and attacks. The proposed approach offers a unified model for agile and resilient distributed computing,

based on standardized technologies for monitoring and sharing of performance and threat data, promising for easy adoption in industry. The proposed performance and security policy enforcement model enables integration of various types of policies and optimization algorithms as well as filtering capabilities (e.g., high-quality vs. lower-quality data) for various data types, which is needed for fine-grain control over dissemination, searches, analytics, and operations in cross domains of privacy.

Future work will include detailed evaluation of the overheads and accuracy of service anomaly detection under various attacks and operational failures as well as extension of the privacy-preserving data dissemination mechanism between the services to a blockchain-based model, where the integrity and validity of the data shared between mission-critical services can be ensured with strong security guarantees.

6. References

[1] J. Simmonds, Y. Gan, M. Chechik, S. Nejati, B. O'Farrell, E. Litani, J. Waterhouse, Runtime monitoring of Web service conversations, IEEE Transactions on Service Computing 2.3 (2009): 223-244.

[2] G. Wu, J. Wei, T. Huang, Flexible pattern monitoring for WSBPEL through stateful aspect extension, Proceedings of the IEEE International Conference on Web Services (ICWS '08), 2008, pp. 577 – 584.

[3] Z. Li, Y. Jin, J. Han, A runtime monitoring and validation framework for Web service interactions, Proceedings of the Australian Software Engineering Conference, Sydney, Australia, 2006, pp. 70–79.

[4] Z. Malik, A. Bouguettaya, Rateweb: reputation assessment for trust establishment among Web services, VLDB 18.4 (2009): 885–911.

[5] G. Spanoudakis, S. LoPresti, Web service trust: towards a dynamic assessment framework, Proceedings of the IEEE International Conference on Availability, Reliability and Security (ARES'09), 2009, pp. 33–40.

[6] R. Xie, R. Gamble, A tiered strategy for auditing in the cloud, Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD), 2012, pp. 945-946.

[7] R. Baird, R. Gamble, Developing a security meta-language framework, Proceedings of the 44th Hawaii International Conference on System Sciences, 2011, pp. 1-10.

[8] Splunk, 2020. URL: http://www.splunk.com.

[9] Graylog, 2020. URL: http://www.graylog.org.

[10] Kibana, 2020. URL: https://www.elastic.co/products/kibana.

[11] S. P. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28.2 (1982): 129–137.

[12] B. Scholkopf, J.C. Platt, J.Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional sistribution, Technical report, Microsoft Research, MSR-TR-99-87, 1999.

[13] R. Ranchal, Cross-Domain Data Dissemination and Policy Enforcement, Ph.D. thesis, Purdue University, West Lafayette, IN, 2015.

[14] R. Ranchal, B. Bhargava, L.B. Othmane, L. Lilien, A. Kim, Protection of identity information in cloud computing without trusted third party, Proceedings of the IEEE International Symposium on Reliable Distributed Systems (SRDS), 2010, pp. 368-372.

[15] P. Angin, B. Bhargava, R. Ranchal, N. Singh, L. Lilien, L.B. Othmane, An entity-centric approach for privacy and identity management in cloud computing, Proceedings of the IEEE International Symposium on Reliable Distributed Systems (SRDS), 2010, pp. 177-183.

[16] B. Bhargava, P. Angin, R. Ranchal, S. Lingayat, A distributed monitoring and reconfiguration approach for adaptive network computing, Proceedings of the 6th International Workshop on Dependable Network Computing and Mobile Systems (DNCMS) in conjunction with SRDS'15, 2015, pp. 31-35.

[17] R. Fernando, R. Ranchal, B. Bhargava, P. Angin, A monitoring approach for policy enforcement in cloud services, Proceedings of the 10th IEEE International Conference on Cloud Computing (CLOUD'17), 2017, pp. 600-607.