Towards a metric-based evaluation of Secure Software Oriented Architecture in Cloud Computing

Clinton Dsouza
School of Computing, Informatics and Decision Systems Engineering
Ira A. Fulton Schools *of* Engineering
Arizona State University
Tempe, Arizona-85281
Email: cvdsouza@asu.edu

***Abstract*—Cloud computing has been contemplated to be a key player in the rising use of mobile and web applications. The uniqueness of the cloud computing paradigm is that it allows for applications to be hosted in a multi-tenant environment. The increased use of cloud computing and it's continuous accessibility expectancy by clients, has driven this paradigm to host a wide array of applications, especially those which require data-intense services. In addition to these services, the cloud computing paradigm can also be utilized to host web services for consumption by developers whilst creating their applications. This multi-usage capability allows for the smooth integration and realization of a software oriented Architecture(SOA) framework within such an elastic paradigm. However with any new framework, there arises questions of security from a multi-dimensional perspective. This paper reviews proposed security frameworks for SOA and evaluates them against well defined metrics to calculate the risk of each framework being hosted in a data-oriented cloud computing environment. The proposed metrics take into consideration both the security criteria for a secure SOA and cloud computing model. The goal is to highlight the risk areas of when a SOA model is combined with a cloud infrastructure and to propose additional criteria that need to be taken into consideration when hosting distributed services in a SOA model in the cloud.**

I. Introduction

The increased use of the "pay-as-you-go" cloud computing paradigm has given rise to a higher accessibility, and availability expectancy from clients utilizing resources in the cloud. This has also opened the cloud to a wide array of application and service hosting capabilities. Apart from hosting sophisticated applications, clients have also realized the usefulness of the cloud computing paradigms to support distributed architectural frameworks such a as Service Oriented Architecture(SOA) which within itself consists a wide range of programmable components which can be developed and hosted by multiple developers. Utilizing the power of the cloud paradigm, developers can develop services for consumption and host them in the Software as a Service(SaaS)layer of the cloud infrastructure and make them available for other developers to use. Similarly, developers could also develop complete web service applications which clients can dynamically embed into their applications. Due to the assurance of on-demand servicing, clients are assured continued access to their applications. Services developed vary based on their intended purpose. Our focus in this paper will be on data-centric applications, that are developed to address application usage in which clients upload large amounts of data to the applications.

The uniqueness of the SOA architecture is its distributive nature and the notion of "separation of duty". Separation of duty in terms of SOA is the concept of developing independent programmable components which can dynamically be consumed by each other when required whilst still functioning independently. The integration of a SOA based architecture along with a Cloud Computing paradigm raises several security issues that need addressing. These security issues are not limited to the cloud but also extends to the architectural framework of a SOA environment.

The Service Oriented Computing(SOC), a distributed computing paradigm is derived from SOA. This architecture describes software systems as a collection of loosely coupled service the have the capability of communicating with each other through standard interfaces [1]. These services are platform independent, implying that they can be hosted on any operating system with a compatible server. It is important to understand the functionality of SOC since all the evaluations conducted on a SOA model, are applicable to SOC due the fact that SOC derives is architecture from SOA.

Tsai et al. in their paper [2] propose an architectural framework for a combination of Service-Oriented Computing paradigm with cloud computing. However their architecture does not discuss the security layers that need to be in place to attempt the building of such an architecture. A simple point to consider as an example is the multi-tenant architecture for a Service Oriented Cloud Computing Architecture(SOCCA). In a multi-tenant architecture, privacy and data integrity is a crucial security criteria to enable secure collaboration between multiple cloud infrastructure. Although SLAs are a designed to facilitate for a multi-tenant client interaction, the architecture fails to account for SLA conflicts between different providers. Further discussions will follow in the coming section to compare similarly proposed architectures and the missing components of consideration which need to be included to ensure seamless and secure communication and collaboration [3].

The primary reason for considering a SOA and cloud computing architectural combination is to propose a mechanism or architecture to handle big data transaction more securely. Big Data can be briefly defined as a concept of bringing together unfathomable amounts of data in a distributed environment to life, in the sense that this data can be mined or searched intelligently to generate meaningful data to assist consumers or clients in their tasks. 2.7 Zetabytes of data exists in the digital world today [4]. This data includes medical information from

the health-care industry which contain highly sensitive patient data which require to comply by HIPAA regulations. This data is utilized to track chronic illnesses, cost analysis and population monitoring of patients in a dynamic cost effective health environment [5]. Data can be categorized as (i) Structured data (eg. lab data, forensic data), (ii) unstructured data (eg. Post-lab results). These categorizations are further classified into imaging data and streaming data. Considering the quantity and density of data and the integrity that needs to be maintained, there are a lot of security concerns with maintaining the confidentiality of user information [5]. Additionally, when data is stored in a cloud infrastructure, and needs to be continuously queried and analyzed for results, there is a requirement for a secure infrastructure to ensure data integrity is maintained throughout all transactions. Given the big data problem we face today and the growing demand to access big data distributively in the cloud. This paper attempts to compare and contrast the security frameworks in place for cloud infrastructures, and SOA frameworks. We utilize the notion of metric to quantify our evaluation and backup our result utilizing the Common Vulnerability Scoring System(CVSS), which serves as a reference guide to the interpretation of our analysis.

Section II discusses the cloud computing infrastructure, including the types of environments developed for deployment. Section III addresses the security challenges in cloud computing across the two main layers of the cloud model. Section IV then goes on to explain the Service Oriented Architecture(SOA) design and framework, along with important concepts that need to be considered thus paving the way for the discussion in Section V. This section discusses the various papers that propose security frameworks in a SOA design. Section VI evaluates the currently proposed SOA frameworks, and their combination with cloud computing thus evaluating the security criteria for data-oriented web services that will be hosted. Section VII interests the results obtained and draws conclusions based on the risk level and the CVSS scores. Section VIII discusses potential future work in ensure a secure SOA in cloud computing following with concluding remarks in section IX.

II. Cloud Computing infrastructure

The concept of cloud computing dates back to the 1950 when main-frame computers were viewed as futures to the computing era. However it wasn't until the early 2000's that the notion and paradigm of cloud computing came to realization and utilization. In this paper we define cloud computing as: a model that enables ubiquitous access to share pool of resources that include computing , storage and networking that can be rapidly provisioned on-demand with minimal management thus enabling convenient and uninterrupted access [3]. Fig 1 provides an overview of the cloud computing infrastructure by showing the different layers that make up the cloud.

The cloud computing paradigm has evolved to a three-layered architectural design comprising of:

- Software as a Service(SaaS): The application layer of the cloud computing paradigm designed to host client-centric applications that can be accessed by users worldwide at anytime.
- Platform as a Service (PaaS): This delivery layer is designed to include platform tools such as application execution environments, operating systems, database and web-servers.
- Infrastructure as a Service (IaaS): The physical entity of the cloud paradigm, which more often also involves a virtual component such as virtual machines running on hyper-visors. The IaaS is designed to provide the physical components required to support the PaaS and SaaS. This includes, server data-centers, hard-disks, hyper-visors, networking routers and more.

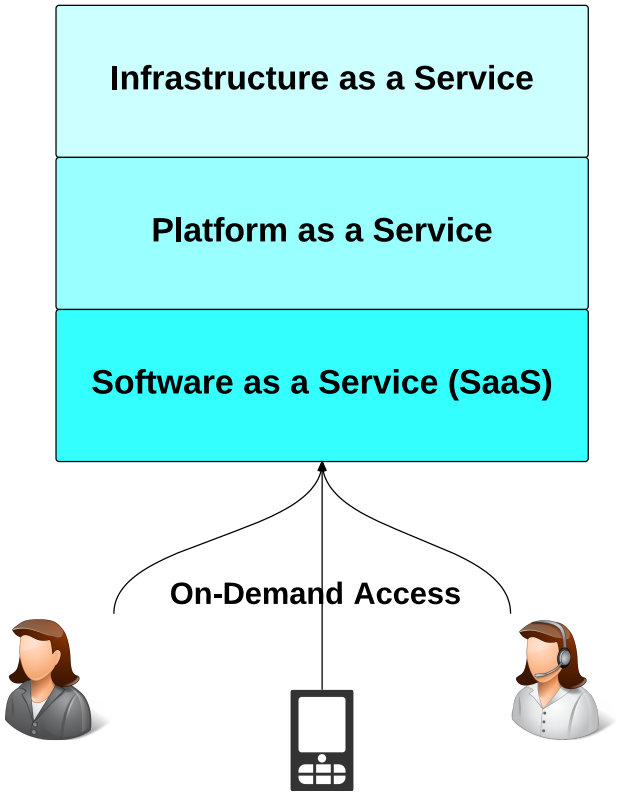

Fig. 1. Cloud computing Infrastructure

The cloud as a whole has evolved into different deployment models based on its usage. The three most common models include:

- Public cloud: designed as an "all-access" model wherein all software and application deployed are accessible to authorized public users who pay for the services being provided.
- Private cloud: designed as an "internal-access" model mostly directed towards enterprise and corporate environments, where employees within a firm are able to access service to assist clients based on their needs. Public access is limited at the discretion of the tenant.
- Hybrid cloud: designed as an "combinatorial-access" model wherein a combination of both public and private cloud models are utilized to enable two-fold architectural access. This deployment model allows public authorized users to access certain services in the cloud but restricts certain applications to only private

cloud users (eg. employees in a firm). This model enables for an easier client-company resource access, as the firm can control service that need to be made public and private.

This paper focuses on two unique combinations of the cloud paradigm. To better analyze the security gaps with combined with a service oriented computing paradigm, we focus on the Software as a Service layer hosted on a public cloud infrastructure. The purpose of this combination if the realization that any SOC architecture design, would prefer to leverage an "all-access" model of deployment to allow for published services to be accessed by authorized users at any time. We make the assumption here that service deployed in a private deployment model are done so by the tenant so as to maintain an "in-house only" access.

Given the choice of model selected for analysis in this paper, our next step is to analyze the security infrastructure protecting this model and the possible flaws that exist or can be potentially be exploited to violate the integrity of collaboration and communication within this model. The next section discusses the security challenges faced when executing applications in the SaaS layer of a public cloud infrastructure.

III. Security challenges in Cloud computing

The maturity of the cloud computing paradigm, implies an improved security infrastructure since its conception. However, with any new evolving technology vulnerability and exploits are imminent.In a public cloud deployment model there are fewer layers of security compared to a private cloud mode. This makes the public cloud prone to a number of attacks, the most common being Man-in-the-middle, botnet attacks, Distributed Denial of Service(DDoS) attacks and more. Since the focus of this paper from a cloud paradigm perspective, revolves around a public cloud and the SaaS layer it is important to understand the underlying infrastructure of the cloud model. Fig 2 gives a hierarchical overview of the critical layers in a cloud infrastructure inclusive of the security measures in place to ensure data integrity, access control assurance and policy management enforcement.

Since this paper will focus on the SaaS and PaaS layer from a cloud computing perspective, we will analyze the security challenges in this particular layer of the cloud.

A. Security challenges in SaaS

The Software as a Service(SaaS) layer of the cloud computing infrastructure is the front line of access through which clients utilize services to which they are provisioned and authorized for. Accessing an SaaS application requires user authentication and authorization wherein a user provides their credentials for verification. Depending on the application type, the application service segregates the user data while in use thus ensuring that data integrity is maintained throughout the access of the service. Once finished access, the service stores the data in a multi-tenant database along with other similar service data. Multi-tenancy in a cloud environment allows for a database to segregate client data so that one service application cannot access client data without prior authorization.

From an SaaS attack perspective, there are two major points of entry attacks:

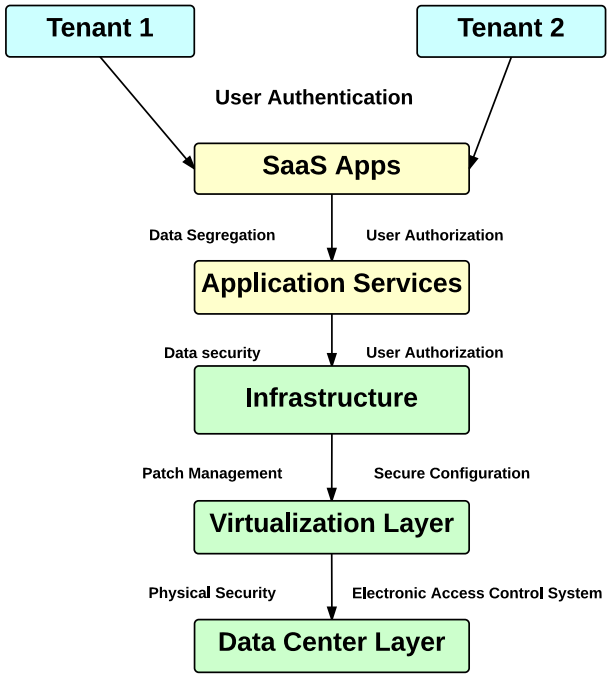

Fig. 2. Security infrastructure in Cloud

- Client-side attack: focuses on direct attack on the web-application.
- Provider-side attack: focuses on exploiting back-door points of weakness into the cloud infrastructure. These could be at the PaaS level such as a web-server vulnerability, or an admin privilege escalation attack targeting a weakness in the SaaS application itself [6].

The challenge to securing the SaaS layer is the attack surface it presents. With traditional systems, an attacker targeted primarily the infrastructure such as the web-service, or the server itself utilizing attacks such as Denial of Service(DoS) or privileged escalation. However with organizations relying heavily on the SaaS layer, attackers are more interested in targeting the access rights of users as data integrity is a key aspect of the SaaS layer. Through the breach in user access right attackers can utilize the SaaS layer to gain access to confidential data which otherwise would require attacking the infrastructure of a traditional system.

B. Security challenges in PaaS

The Platform as a Service(PaaS) layer of the cloud computing paradigm can be referred to as the "developmental layer" among the three primary layers. The PaaS layer is responsible for providing a platform for developers to develop their application upon.The tools provided are in the form of web-servers, operating systems, and programming language sdk's. The scenarios of the PaaS layer are similar to those discussed in the SaaS layer above. There are different types of PaaS offerings, the important ones include:

- Add-on development: They are customizations of the

SaaS applications to behave as packaged software applications.

- Stand-alone development : provides generalize development environment such as the basic server instance but doesn't provide development , technical, licensing or financial specific environments.
- Application delivery-only: they do not provide development, debugging and test capable services.
- Open platform as a service: provides open source software to enable a PaaS provider to run application based on service needs.

From a security standpoint , the PaaS serves as a building block to developing a secure application. All applications in a cloud infrastructure utilize the tools hosted on a PaaS layer of the cloud, hence the security infrastructure requires fine grained analysis and design. Key points of interest in the PaaS layer is the data security, user authorization, patch management and secure configuration.

C. Cloud Computing and Service Oriented Architecture

Service Oriented Architecture(SOA) and cloud computing are inter-linked with each other. The SOA architecture is designed for distributed systems supporting creation, organization and reuse of computing components. Cloud computing is a paradigm designed to enable provisioning of flexible resources and platforms for enterprise organizations to build their SOA architecture upon [2]. The cloud computing model can be viewed from two perspectives: (i) User(clients and tenants), (ii) Developers. This paper will focus on cloud computing from a developers perspective since SOA is focused on the development life cycle of independent computing components designed as services. The unique difference between SOC and SOA is that SOA is a conceptual model and doesn't depend on algorithmic design and implementation to create operational software [1].

Tsai et al. in their paper of Software Oriented Cloud Computing Architecture propose a layered architecture for SOCCA designed to combine the functionalities of a SOA architecture and a cloud computing model.Utilizing the concept of ontology mapping they proposed a cloud ontology mapping layer of which further analysis and comparisons will be made from a security perspective in chapter VI. Finally, cloud computing and SOA overlap over certain of points of interest which will better help in understanding the motivation behind our evaluation:

- Both SOA and cloud contain a application service layer. While the cloud has three unique layers inclusive of the SaaS layer, the SOA primarily focuses on the application layer.
- Both SOA and cloud revolve around a produce/consumer model.
- They both depend on a IP/wide area network(WAN) support for service access and invocations.

IV. Service Oriented Architecture design

Based on a Service Oriented Architectural framework, a Service Oriented Computing system will comprise of independently developed modules with little to no interlinked

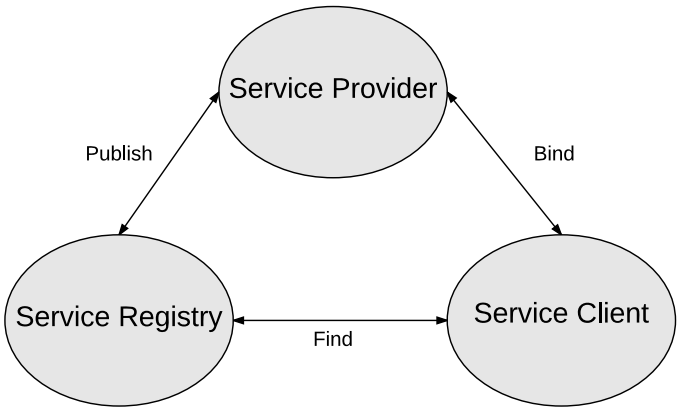

Fig. 3. Basic SOA

services.The focus of SOC approach is based on the service definition and the predictability of the service behaviors [7]. Papazoglou et al. in their paper [8] propose three requirements that need to be satisfied for services in a Software Oriented Architecture: (i) Technology neutral, (ii) Loosely coupled and (iii) Support for location transparency. The services developed in this architecture are classified into two flavors; simple and composite services.

The basic components of a SOA framework are:

- Service Provider: Software agents that provide services. These services can be bound to the client or can be published to the service registry.
- Service Registry: Provides a registry of services that a client can request binding to.
- Service Client: Software agents (web applications) that request execution of services through the registry.

Fig 3 displays a basic SOA model upon which a SOC framework is built upon. Each component in this model is designed to interact with each other so as to provide the end-user with optimal services. As iterated earlier the services are independent thus allowing for heterogeneous services to communication with the service client.

Services can be defined as self-contained units that perform a certain functionality. These could be varying depending on the service owner [9]. Our paper focuses on data-oriented services that involve large data transfers, migration and analysis. These web services are basically SOA based architectural models applied to data sourced and stored both from online sources and from users, they are commonly referred to as Web data services.

A. Extensions of Service Oriented Architecture

The maturity of the SOA framework, has led to researchers such as Papazoglou et al. to propose state of the art extended SOA frameworks [8], that takes into consideration more fine-grained service interface components that can be built and reused based on service requirements. The extended SOA(ESOA) architecture extends the basic SOA framework into two layers: (i) Service-deployment, (ii) Service-realization. This layered architecture utilizes the basic SOA

architecture in Fig 3 as it's bottom layer. The highlights of this architectures is the service aggregators which perform functions including: Coordination, Monitoring, Conformance, and Quality of Service(QoS).
As discussed in earlier sections, Tsai et al. in their paper [2] proposed a service-oriented cloud computing architecture that can interoperate between the cloud and SOC infrastructures. This multi-tenant architecture is well developed to support several cloud computing organizations(Amazon EC2, Microsoft Azure, Saleforce) utilizing a cloud broker service and ontology mapping to map application instances (web services) to their respective resource (storage, computing and networking)in the designated cloud infrastructure.
Dikanski et al. in their paper [10] propose a view-based approach for service oriented security architecture. This paper proposes SOA security model incorporating the different views of security engineering, security infrastructure and software engineering development models.

However none of the above extended architectures focus on the security infrastructure for handling big data oriented services but rather focus on the mapping of services. Further analysis of SOA frameworks with security components are discussed in the next section.

V. Security evaluations of current proposed SOA framework

This section will discuss, compare and contrast the unique and prevalent security architectures for SOA. We will additionally, discuss the security criteria from a data security perspective when the discussed SOA are combined with a cloud infrastructure. This section and its subsequent subsections will discuss each architecture and its contributions to the generic SOA model.

Before discussing the security evaluations of SOA and cloud frameworks, we establish what security is in terms of cloud computing and SOA. The term computer security has evolved in the recent years to cover a lot of different areas in technology. However to specifically understand what security is when referred in cloud computing and SOA we refer to it as follows: Protection of data and systems from "unauthorized access, use, disclosure, disruption, modification or destruction" [11] so as to maintain the confidentiality, integrity, and availability of systems and data all times. The following subsections will discuss three unique approaches to security in a SOA model. The purpose of the discussion of these approaches are to help the reader better understand the evaluation criteria that will be defined later in this paper to evaluate the security risks posed by SOA and its integration with a cloud infrastructure.

A. Secure Audit-based approach for Service Oriented Architectures

Azarmi et al. in their paper [12] claim to provide an efficient solution for secure auditing in SOA. They introduce two sub-systems into their architecture which will be shortly discussed. These components take advantage of WS-security and WS-Trust standards to ensure secure transactions in their SOA architecture.

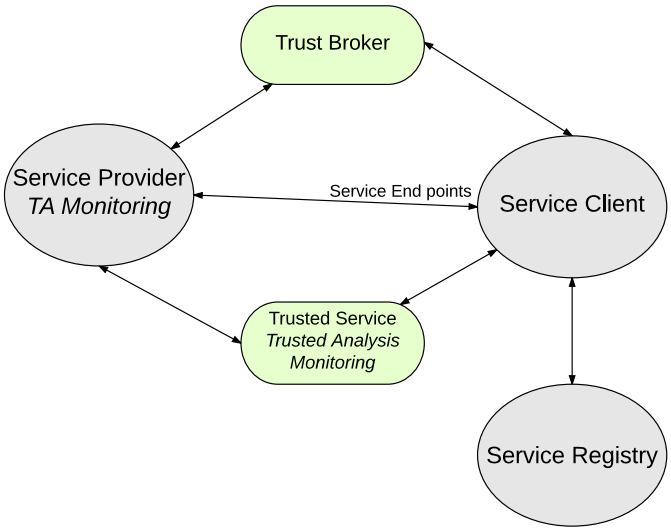

Fig. 4. Basic SOA with secure auditing components

1) Discussion: An audit system in general is referred to as a monitoring system wherein developed modules monitor activities of service in a system and detect suspicious activities. [12] proposes a *Taint-analysis(TA)* system which is a auditing system, to keep track of activities of services at run-time, and inspect data-exchanges to detect anomalous events. The taint analysis sub-system is designed to function at run-time thus serving as a continuous monitoring system. When an anomaly is detected the TA system takes control of the program running the anomaly to determine the cause and correction. To achieve this the authors propose utilizing Aspect Oriented Programming (AOP) as language for special information flow control for when auditing the system in question.

The second component proposed in addition to the TA system is the *Trust Broker*(TB) System. This system is designed as a third party system added to a SOA with the functionality of maintaining end-to-end security in a chain of service invocations deployed in a SOA framework. These interaction are primarily between the clients and the associated services. The TB perfoms the following major functionalities:

- Maintaining list of certified services.
- Maintaining end-to-end sessions of service invocations.
- Evaluating the trust level of service.

Fig 4 depicts a high-level view of the interaction of the proposed monitoring system. The figure clearly shows the interaction of the monitoring components and their placement in the basic SOA framework.

2) Contributions: This proposed architecture provides an end-to-end security solution for SOA utilizing auditing as a mechanism to monitor and resolve anomalies in a distributed system. They introduce two relatively unique concepts in security namely: Trust Broker Service and Taint Analysis monitoring. Through the ability to monitor events in a module

occurring at run-time, the proposed solution attempts provide dynamic real-time security in an SOA framework.

B. View-based approach to secure Service Oriented Architecture

1) Discussion: When analyzing a security architecture, there are varying perspectives that need to be taken into considerations. However if an enterprise already has a secure SOA model, and more components need to be added, there will be certain measures that need to be considered so as to avoid breach in data. Dikanski et al. address these possible security consideration that need to be taken into consideration by making changes to an existent secure SOA. The different perspectives in viewing a security architecture assist researchers in analyzing the security architecture of a proposed infrastructure and viewing them from different angles.Dikanski et al. in their paper [10] propose a view-based approach for secure SOA specifications. They propose three architectural views which provide details about the specific components their purpose in the over SOA framework. The three views proposed are:

1) Security Engineering Views: Focuses on the information that a SOA provides. More specifically the focus lies up the service analysis and discovery including service design. There are three main phases that build up the security engineering view. These include :
   - Security Requirement Analysis : whose goal is to specify requirements for an application that is utilizing the secure SOA.
   - Security Design Pattern: are specified as part of the design architecture so as to keep them separate from the application design. The design patterns take into account all the requirement analysis, measured and specify them with relation to previous set security requirements.
   - Security Policy Model: policies are utilized to control the behavior of components within the architecture. The primarily focus upon the behavior of security service and application being utilized with the described SOA model.
2) Security Service View: this components are concerned with provisioning of secure services for the defined secure SOA. The service view consists of a set of specific guidelines and principles with ensure secure mapping of all standards and protocols to their respective services. The security service view comprises of two components:
   - Service Interface and Protocols
   - Security standards and guidelines.

   Since the terms are self explanatory, we will not go into more details in this paper.
3) Security Integration View: when a re-structure of an existent SOA model takes place, it is required to have a centralized component to integrate the new service with the existent one. This view takes care of the integration. The security integration view consists of three components:
   - Security Service Discovery
   - Service-Oriented Integration
   - Security Service Infrastructure

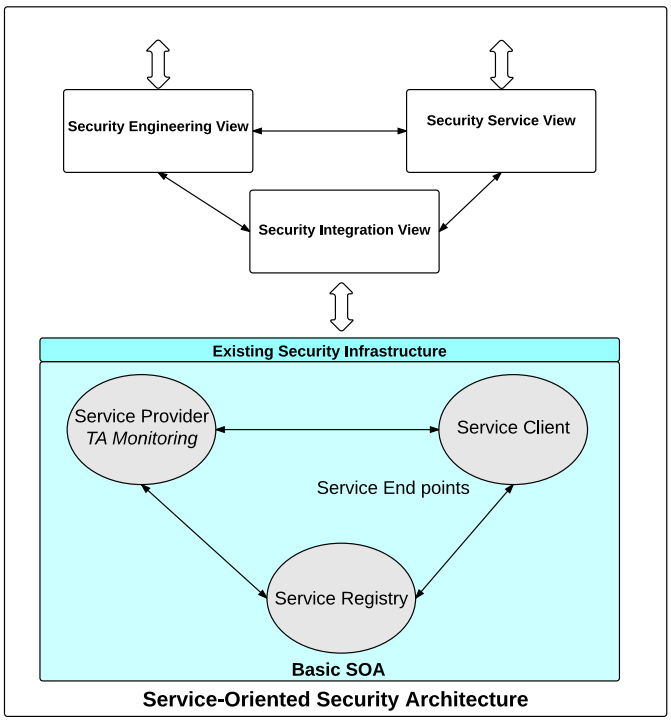

Fig. 5. Secure SOA model combined with basic SOA

In addition to the security architecture, this paper proposes an access control service specification for maintaining authentication and authorization among the different components in the secure SOA. They propose utilization of Role-Based Access Control(RBAC) as the primary access control model in a web service specification. Fig 5 depicts an abstract view of the proposed secure SOA model.

2) Contribution: The primary contribution of this paper is the presentation of a service-oriented security architecture mode based on a view-based approach. Each view provided an interensic view into the different security consideration that need to be considered in a SOA model. Finally, they proposed the utilization of an access control model(RBAC)to manage the access privileges of the different views in a web service specification.

C. Data-mining as a technique to secure SOA

1) Discussion: Data mining is the process of analyzing data from different perspectives and then summarizing it to produce intelligent results that users can make sense of and utilize [13]. Data mining has a long history of supporting a lot of computing services, however with the increasing use of web-service in SOA models, data-mining is being considered as useful technique in securing the distributed services that run within a SOA model. Yamany et al. in their paper [13] propose a security service for a basic SOA framework utilizing data mining as a methodology to secure the communication between the different services running within the framework.

The primary structure of the proposed secure SOA model

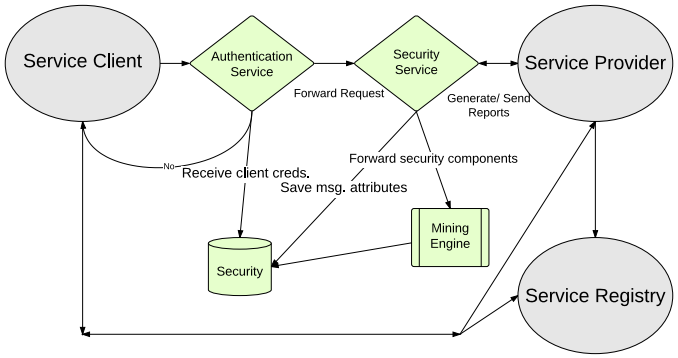

Fig. 6. Secure SOA model combined with basic SOA

utilizing data-mining can be broken down into the following steps:

1) Service consumer sends a SOAP message requesting for access to services hosted by the service provider.
2) Authentication service on the provider side validates the identity of the requester.
3) Intelligent Security Service (core component) processing the incoming message request(SOAP) inside a database and extracts the message attributes:
   a) Message(SOAP or REST) is parsed
   b) Security features from the message is stored into the built-in mining model.
   c) Analyses of the extracted messages from the mining model could lead to prediction of attacks.
   d) Based on available data from mining model classify service clients/ consumers.
4) Generate report for service provider and respond to consumer with authentication result.
5) Security administrator tests a new policy prior to deployment. Test determines the validity of the policy.

Fig 6 depicts the workflow of the proposed security service utilizing a data-mining engine.

2) Contribution: The primary contribution of this paper is the proposal of a new security service that attaches to the existent SOA model thus enabling for a more secure SOA infrastructure. Additionally, this service utilized a data-mining analysis model that allow for deep data analysis and attack prediction utilizing message attributes extracted from the stored messages.

VI. Metric analysis, evaluation and results

A metric is a measure of a particular characteristic of a program or algorithm's performance or efficiency [14]. Metrics enable researchers to evaluate one or more models or program and deduce it's differentiating characteristics and uniqueness. This paper utilized the same characteristics of metrics but in a different way. Rather than evaluating performance, we evaluate the security criteria of introducing a secure SOA model with a public cloud infrastructure given the cloud infrastructure is designed to orient towards handling data services. Our evaluation focuses on the security gaps that have not been considered, and those issues of which without resolution, the data integrity has a high potential of being violated or breached.

A. Metric definitions

Our evaluation is based on the intersection of metrics from two different models: (i) Public data-oriented cloud computing, (ii) Secure Service Oriented Architectures. These metric will be applied to three research approaches that were discussed in this paper: Section V-A, Section V-B, Section V-C. Each paper proposed a unique security framework to enable a secure SOA model. However as iterated in section III, cloud computing as a stand-alone infrastructure does have certain security challenges. We thus define metrics for both the cloud and SOA model so as to realize the challenges that will be faced when secure SOA based services are hosted in a cloud infrastructure.

1) Cloud computing security metrics: Measuring the security in a cloud infrastructure is a challenge due to its multi-level architecture with varying programmable components. However since our focus in on measuring and quantifying the security challenges from data-oriented infrastructure, we classify the metrics into three primary categories:

1) Confidentiality: Data confidentiality measures preservation of authorized restriction on information access [11]. This is measured through the extent of sensitive data accessible to a malicious user during or after a breach in systems occurs.
2) Integrity: Integrity of data is measured through guarding against unauthorized and improper modification or destruction of information whilst ensuring information non-repudiation [11]. Integrity of data is measured through the activation of counter security or back-up measures protecting the sensitive data when a violation in policies or access control mechanisms occur.
3) Availability: Data availability is measure in *time*(hours, minutes, seconds). Availability ensures reliable and timely access to user information and prevention of disruption to services [11].

2) Service Oriented Architecture security metrics: Security in Service Oriented Architecture has to go beyond the basic fundamentals (Confidentiality, Integrity, Availability, Usage) of computer security. SOA models involve distributed systems, with independent components. Thus inclusive of the basics fundamentals we propose three metrics based on which a secure SOA model can be measured:

1) Loose coupling: SOA service are designed to behave independently, without cross-service dependencies. However, they do require to communicate with each other based on the design of the service.SOA model uses SOAP protocol as the most common mechanism in specifying messages since it uses XML to send them. Loose coupling metrics analyzes the message transmission and based on transmission type, security measurements can be determined in based on the potential for message integrity violation.
2) Discoverability: SOA models promote service discoverability where service allow their services requestors

to discover them and make request for use of their operations [15]. Discoverability metric, measures the extent to which a service is vulnerable by making itself discoverable. The metric is measured based on maintenance of service integrity and continuity in the event of a vulnerability discovery.

3) Interoperability: SOA models promote interaction with each other and other related models. As a consequences, security risks arise from third party interactions. This metric measures the impact of a third part compromised service on a give secure SOA model.

Our approach is to combine both metrics, and measure each approach of a secure SOA architecture against the SOA metrics at the same time measuring them against the cloud infrastructure. This will allow us to view any security gaps or criteria that we have to consider.

B. Evaluation

1) Secure Audit-based approach for SOA (Section V-A): The audit-based approach discussed in section V-A refer to two main systems: Taint-analysis(TA) monitoring system, and Trust Broker(TB) system. Based on the metrics defined above, we attempt to quantify the risk of exposure through classification into three categories : High Risk(H), Medium Risk (M) and Low Risk (L). Table shows the overall classification and metric measurement. Following are the deductions:

1) Loose Coupling (H): As observed from 4 the two systems are designed to enhance security , however they only communicate with a verified service. Since the SOA framework encourages loosely coupled service, not all services are verified. Additionally, a service monitor with a TA monitoring sub-system can still connect directly to the service client. This implies that although monitoring is enabled, the client credentials are not verified by the Trust Broker before usage.
2) Discoverability(L): Although the paper makes no specific deductions on discovering of services, the TB is designed to verify user credentials before making connections. Thus if a user is attempting to discover a secure service, he will have to go through the TB system for credential validation. Thus not all service are discoverable to the public.
3) Interoperability(H): This approach doesn't specify any interopereability capability of the proposed architecture. However, given that the architecture is that of a simple SOA, there are no provisions on handling third party un-verified systems thus putting the system at high risk from interoperability.
4) Confidentiality(L): The TB system helps maintain a list of certified service thus clearly enforcing confidentiality which is vital in a cloud infrastructure, since uniform confidentiality require trust to be established between a service and its hosting virtual instance..
5) Integrity(L): The TA system monitors the service providers of which it is a part of, thus ensuring integrity of the data and services being monitored.
6) Availability(M): The architecture proposed doesn't discuss and depict any redundancy or back use-case scenarios. However, the TB and TA subsystems are designed as independent system which can be easily replaced if compromised.

2) View-based approach to secure SOA (Section V-B): The view-based approach discussed in section V-B proposes three security views : Security Engineering View, Security Service View, and Security integration View. Each view accomplishes certain security functionalities which are discussed in the mentioned section. Similar to the above section the risk lever as classified as High, Medium, Low.

1) Loose Coupling (M): The approach doesn't take into consideration communication with other infrastructures, however the design of the system allows for secure integration of other SOA architectures into the proposed secure architecture thus indicating certain level of tolerance to third party infrastructures.
2) Discoverability (L): The proposed approach has a very low risk of discoverability due to the Security Integration View which consists of a service-oriented discover module programmed specifically to allow discovery of secure service alone.
3) Interoperability (H): The proposed approach only takes into consideration other secure SOA architecture like itself, but fails to account for third party service or infrastructure that are not validated, but would still be preferred to be utilized by clients. This raises the risk level.
4) Confidentiality (L): Since the proposed architecture covers the basic fundamentals of security , it encapsulates the metric for confidentiality through all three view combined.
5) Integrity (L): The Security Engineering view and security service view combined ensure that integrity metric is kept at a low risk. The Analysis, Policy model and design pattern present in the security engineering view ensure that data integrity is enforced in the system.
6) Availability (M): Due to the inter-dependency of all views with each other the breach in a single view can bring down the system.

3) data-mining as a technique to secure SOA (Section V-C): Utilizing data-mining as technique in section V-C a secure SOA is proposed. The proposed approach utilized a security service connected with a mining engine to authenticate users and extract useful attributes from messages send between clients and providers for analysis.

1) Loose coupling (L): The proposed approach utilizes the basic SOA model thus allowing for loosely coupled services to be added or accessed by clients on demand. Given the security structure as depicted in 6 all client are validate prior to access to any services.
2) Discoverability (M): The approach proposed has no discoverability restrictions, which means that any service provider's services are discoverable to the public.

Approach \ Metrics	Loose Coupling	Discoverability	Interoperability	Confidentiality	Integrity	Availability	CVSS Score
Audit-based	H	L	H	L	L	L	6.8
View-based	M	L	H	L	L	M	5.3
Data-mining-based	L	M	H	L	H	M	8

TABLE I. METRIC EVALUATIONS

However given the security flow of the approach, the risk of exploitation is medium due to the fact that a user has to be authenticated prior to accessing a service.

3) Interoperability(H): Since the proposed approach utilizes the basic SOA model with intermediate security measures, interoperability poses a high risk to revealing sensitive information of a SOA model. Since each service provider has no security measure within itself, communication with third part service providers could potentially create data integrity violations.
4) Confidentiality (L): Given the client-provider intermediary security modules, breach in confidentiality is of low risk due to the presence of Authentication service and security service modules.
5) Integrity (H): The service providers having no security modules, embedded in them, thus the integrity of data can be validated when passed through a service provider.
6) Availability(M): Availability is measured by the access to data from the service provider and the security database in the event of an attack. A medium risk indicates that the security database will be accessible however the service provider data hasn't been secured well enough due to a regular SOA infrastructure.

VII. RESULTS

Based on the thorough evaluation conducted in the section above, Table I provides an overall view of the metric evaluation against the proposed approach for building a secure SOA infrastructure in a cloud environment. The metric evaluations are classified into High Risk(H), Medium Risk(M), and Low Risk(L). These classifications enable us in deducing the risk levels of each metric compared to their approach. Since our proposed metrics are preliminary, we utilize a *Common Vulnerability Scoring System(CVSS)*to support our claims. The scoring system follows a Base Score system that reflects the "intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments" [16]. To better interpret the scoring system we first take a look at the six fundamental metrics: Access Vector, Access Complexity, Authentication, Confidentiality Impact, Integrity Impact, and Availability Impact. Based on our interpretation of our defined metrics, we enter the scoring system into CISCO IntelliShield system which provides a base score based on the entered parameters.

Based on the CVSS system if the base score is greater than 7.5, the compromise is down to the OS of the targeted system. Subsequently, a value less than 7 would indicate a more effective system. Based on the CVSS scoring system view-based based approach for service-oriented security architecture [10].

The view-based approach utilizes a combination of secure engineering, software engineering and software security fundamentals to create a secure SOA model. Due to this, the approach has the capability to cover certain fundamental security concepts required for cloud computing. However all three models have a high risk of interoperability. In a distributed environment such as SOA, there is a need for a distributed system where each component is independent. If this system is then loaded on to a centralized system such as cloud but still distributed in nature, there is a high possibility that those risks in an interoperable system will compromise the data store in a cloud computing infrastructure.

To truly understand and realize a secure data-oriented cloud infrastructure hosting a secure SOA framework, the following primary propositional points needs to be followed in addition to those mentioned in the metrics:

1) Storage Broker: Data being a prime entity of consideration in this research, needs to have a broker so as to efficiently allocate storage space in a cloud infrastructure based on provisioning in a SOA architecture. SOA being a distribute architecture, requires a mature storage mapping to dynamically allocate storage to VM's in the cloud on-demand.
2) Policy mapping: To realize the power of the cloud, a single cloud infrastructure is not sufficient. Enterprise level architectures involve multiple cloud infrastructures communicating with each other. This requires a uniform policy analysis model that is able to map policies from different cloud computing infrastructures to a uniform language specification. For example: XACML (Extensible Access Control Markup Language) is a specification language that monitors polices and rules and helps define a uniform policy specification schema.
3) Computing Broker: A computing broker sits at the edge of the cloud infrastructure and a service oriented computing system. The broker maps the computation environment to its respective SOA service component. Having the capability to launch multiple virtual instances from the hyper-visor and allocate resources on-demand, the broker makes sure that service are allocated the required services on-demand.
4) Access Control Model: We consider access control to be an essential component building a secure cloud infrastructure and a secure service oriented architecture. Based on our evaluation of all approaches, Role-Based Access Control(RBAC) was the prevailing model utilized to maintain access in the system based on a user roles. However in a distributed architecture inclusive of the cloud infrastructure, RBAC is a restrictive model where dynamic role allocation is not possible. Thus a new model is being proposed: Attribute-based Access Control(ABAC). ABAC utilized the notion of attributes to realize the access provisioning of a user on demand [17]. Rubio et al. in their paper [17] provide a formal definition of ABAC

in-addition to defining the security provisioning of attributes in a system. Utilizing ABAC a system can be designed to uniformly provision resources based on a users access credentials(attributes).

VIII. Future Work

The future of a secure cloud computing service oriented architecture needs a more mature model. Based on the suggestions provided in the section VI-B2, a more robust and dynamic model can be created to encapsulate the security gaps between both cloud computing and service oriented computing. Addressing these gaps mentioned in the metrics and the results will assist in the development of a more mature computing model capable of robust distributed computing.

IX. Conclusion

Security is a vital component to any system. Integrity and confidentiality are the key criteria ensure secure data storage, communication and migration. This research evaluated three different approaches to a secure SOA architecture. We discussed the secure infrastructure of a data-oriented public cloud computing environment and design of a service oriented architecture model. The approaches proposed in the evaluated papers were then measured for their risk level based on defined metrics for both cloud computing security and SOA secure models. Finally, to prove the effectiveness of our evaluation we calculated the CVSS score of the base system of each proposed model and proposed a viable solution to ensure a secure SOA model in a secure cloud infrastructure in a data-oriented environment.

Acknowledgment

The author would like to thank Dr. Janaka Balasooriya for his support and guidance in writing this paper.

References

[1] Wei-Tek Che, Yinong; Tsai. *Service-Oriented Computing and Web Software Integration from Principles to Development*. Kendall Hunt, 4050 westmark Drive, dubuque, IA 52004-1840, third edition, 2013.

[2] Wei-Tek Tsai, Xin Sun, and Janaka Balasooriya. Service-oriented cloud computing architecture. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, pages 684–689. IEEE, 2010.

[3] Peter Mell and Tim Grance. The nist definition of cloud computing. 2011.

[4] Heather Taylor. Managing the flood of big data: infographic. IBM Blog, may 2012.

[5] Walt Disney. Big data and cloud computing.

[6] Rafael; Dasgupta Partha Dsouza, Clinton; Santana. Vulnerabilities in cloud computing. This work was part of the Fulton Undergraduate Research Initiative(FURI) at Arizona Sate University. This paper was presented at the Society of Hispanic Professional in Engineering in 2012 at the National Conference in Anaheim, California., nov 2012.

[7] Geoffrey Raines. Cloud computing and soa. *Systems Engineering at MITRE, Service-Oriented Architecture(SOA) Series*, 2009.

[8] Mike P Papazoglou and Willem-Jan van den Heuvel. Service-oriented computing: State-of-the-art and open research issues. *IEEE Computer. v40 i11*, 2003.

[9] The Open Group SOA Working Group. Service oriented architecture : What is soa? Open Group, jan 2013.

[10] Aleksander Dikanski and Sebastian Abeck. A view-based approach for service-oriented security architecture specification. In *ICIW 2011, The Sixth International Conference on Internet and Web Applications and Services*, pages 207–213, 2011.

[11] Vic (J.R) Winkler. *Securing the Cloud*. Syngress, 2011.

[12] Mehdi Azarmi, Bharat K Bhargava, Pelin Angin, Rohit Ranchal, Norman Ahmed, Asher Sinclair, Mark Linderman, and Lotfi Ben Othmane. An end-to-end security auditing approach for service oriented architectures. In *SRDS*, pages 279–284. Citeseer, 2012.

[13] H. Yamany and M.A.M. Capretz. Use of data mining to enhance security for soa. In *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*, volume 1, pages 551–558, Nov 2008.

[14] http://whatis.techtarget.com/definition/metric, 2005.

[15] Varvana Myllärniemi. Security in service-oriented architectures: Challenges and solutions. 2007.

[16] Eric P. Maurice. Understanding the common vulnerability scoring system (cvss). *The Oracle Software Security Assurance Blog*, apr 2011.

[17] Carlos E Rubio-Medrano, Clinton D'Souza, and Gail-Joon Ahn. Supporting secure collaborations with attribute-based access control. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, pages 525–530. IEEE, 2013.