Collaborative Intrusion Detection in Federated Cloud Environments

Áine MacDermott, Qi Shi, and Kashif Kifayat

PROTECT: Research Centre for Critical infrastructure Computer Technology and Protection
School of Computing and Mathematical Sciences,
Liverpool John Moores University,
Liverpool, L3 3AF, UK
a.mac-dermott@2008.ljmu.ac.uk and {q.shi, k.kifayat}@ljmu.ac.uk

Abstract: Moving services to the Cloud is a trend that has steadly gained popularity over recent years, with a constant increase in sophistication and complexity of such services. Today, critical infrastructure operators are considering moving their services and data to the Cloud. Infrastructure vendors will inevitably take advantage of the benefits Cloud Computing has to offer. As Cloud Computing grows in popularity, new models are deployed to exploit even further its full capacity, one of which is the deployment of Cloud federations. A Cloud federation is an association among different Cloud Service Providers (CSPs) with the goal of sharing resources and data. In providing a larger-scale and higher performance infrastructure, federation enables on-demand provisioning of complex services. In this paper we convey our contribution to this area by outlining our proposed methodology that develops a robust collaborative intrusion detection methodology in a federated Cloud environment. For collaborative intrusion detection we use the Dempster-Shafer theory of evidence to fuse the beliefs provided by the monitoring entities, taking the final decision regarding a possible attack. Protecting the federated Cloud against cyber attacks is a vital concern, due to the potential for significant economic consequences.

Keywords: *critical infrastructure; Cloud computing; Cloud federation; collaboration; intrusion detection; dempster-shafer; fusion algorithm; OPNET.*

1. Introduction

Cloud Computing is being adopted in critical sectors such as energy, transport, and finance. This makes Cloud Computing services critical in themselves. Cloud Computing is a model in which vast quantities of computer resources are used to provide services to many concurrent users. The services may be offered directly or as part of a composite system. The greater scalability and larger size of Clouds compared to traditional service hosting infrastructure, involve more complex monitoring systems, which have to be scalable and robust. Therefore, monitoring systems and intrusion detection systems (IDSs) must be refined and adapted to different situations in Cloud environments.

To embrace this challenge, we propose a methodology that develops a robust collaborative IDS in a federated Cloud environment. Our approach offers a proactive collaborative model for Cloud intrusion detection based on the distribution of responsibilities. The responsibility for managing the elements of the Cloud is distributed among several monitoring nodes. Our architecture consists of four major entities: the Cloud Broker, the Monitoring Nodes, the Local Coordinator (Super Nodes), and the Global Coordinator (Command and Control server: C2). For collaborative intrusion detection, we use the Dempster-Shafer theory of evidence. Dempster-Shafer is used to collect and fuse the beliefs provided by the monitoring entities. Collaboration among Cloud Service Providers (CSPs) can ensure that they are up to date on different Cloud threats. Our current work focuses on the deployment of such a solution for CSP collaboration: Security as a Service.

The structure of this paper is as follows: Section 2 provides background on the research problem we have identified, namely the critical infrastructure and Cloud computing progression, and the associated benefits and vulnerabilities of Cloud federations, and the Big Data connection. In Section 3 we discuss related work in the area and present our analysis. In Section 4 we outline our collaborative intrusion detection methodology for federated Cloud environments, "Security as a Service". Section 5 details Dempster-Shafer theory of evidence and its involvement in our decision making process. In Section 6 we convey our implementation details, and present our conclusions and future work in Section 7.

2. Background

2.1. Critical Infrastructure and Cloud Computing utilisation

As more sectors utilise Cloud-based services in their computing environment, Critical Infrastructure services are

likely to adopt this paradigm. Utilising Cloud Computing within an environment that historically has not had any Internet connectivity would appear trivial to some, however research has shown that Cloud Computing will reach the Information and Communication Technology (ICT) services that are operating critical infrastructure [1]–[4]. Operators of critical infrastructures, in particular the ICT that supports gas and electricity utilities and government services, are considering using the Cloud to provision their high assurance services. This is reflected in a white paper produced by the European Network and Information Security Agency (ENISA) in 2011 [2], which provides specific guidelines in this area, highlighting the technical, policy and legal implications.

Cloud Computing can be conveyed as the next logical progression within the critical infrastructure industry as the Cloud paradigm is already being used for crucial assets. The increasing flexibility and unpredictable usage of such utilities often means that many challenges such as load balancing can occur in the utility networks we use. The usage of modern ICT systems to control and manage critical infrastructure helps in dealing with such issues [3]. Many operators do not have the infrastructure to support the growing need for accurate predictive and historical simulations imposed by the adoption of renewable energy sources and the on-going development of smart grids. To overcome this, Cloud Computing allows these operators to reduce or avoid over investment in hardware resources and their associated maintenance [5].

In November 2012 seven of the world's leading telecommunication network operators selected the European Telecommunications Standards Institute (ETSI) to be the home of the Industry Specification Group for NFV (Network Functions Virtualisation). ETSI ISG NFV (ETSI Industry Specific Groups for Network Functions Virtualisation) [6] was formed with the purpose of developing pre-standards for moving telecommunications functions to the Cloud Computing environment. The NFV ISG's mission is to facilitate the industry transformation and development of an open, interoperable, ecosystem through specification, implementation and deployment experience. Other critical infrastructure service operators from the traffic and transportation, and infrastructure surveillance systems domain are expected to follow soon. The promised advantages do not only relate to cost reductions and increased flexibility, but also new ways to improve the resilience and availability of the critical infrastructure, e.g., through the use of abundant virtual resources [4].

An industry who could benefit from this application is the UK energy community as Cloud Computing can address at least two fundamental requirements. Firstly, accurate network simulations require highly variable quantities of computational resources depending on the contingent situation of energy delivery or on the type of energy delivered. Renewable energy output is typically much less predictable than the constant output offered by conventional generation sources, such as coal, oil, gas, or nuclear. For this reason, running simulations on the Cloud allows for dynamic scaling of the required computational and data resources [1].

Deploying high assurance services in the Cloud increases cyber security concerns, as successful attacks could lead to outages of key services that our society depends on, and disclosure of sensitive personal information. However, this exposes these infrastructures to cyber risks and results in demand for protection against cyber attacks, even more than traditional systems. Security is a major concern in Cloud adoption. Critical security issues include data integrity, user confidentiality, and trust among providers, individual users, and user groups. Additionally, availability issues and real world impact would be the main concern for providers of critical infrastructure, depending upon the operations or services they are hosting [7]. There are security issues at each level of the Cloud Computing paradigm. Nonetheless, utilising the Cloud environment is a natural extension of remote access as it removes the requirement for the user to be in the same location as the infrastructure which is already commonplace.

Critical infrastructure imposes much stronger requirements for security, reliability, and resilience on Cloud Computing environments. Issues also surround data being exchanged across multiple countries that have different laws and regulations concerning data traversal, protection requirements, and privacy laws. Examples of such risks include, but not limited to, risks resulting from possible changes of jurisdiction and the liability or obligation of the vendor in case of loss of data and/or business interruption [8]. As evident, their connexion will provide many benefits in the form of scalability, improved performance, reachability, and will be cost effective for organisations and infrastructure vendors, however the distributed and open structure of Cloud Computing and services becomes an attractive target for potential cyber-attacks by intruders. Despite security issues slowing its adoption, Cloud Computing has already become a persistent force; thus, security mechanisms to ensure its secure adoption are an immediate need.

2.2. Cloud Federations

Cloud Computing hides resource availability issues making this infrastructure appealing to users with varying computational requirements: from storage applications to intensive computing tasks. Large-scale parallel simulations often require computational time on high performance computing machines and clusters. In a Cloud Computing environment resources are shared among multiple users. The number and nature of the workload presented by these users can vary over time. As Cloud Computing grows in popularity, new models are deployed to exploit even further its full capacity. One of these ideas is the deployment of Cloud federations.

Federated Clouds are a logical evolution of the centralised approach. They involve multiple Clouds that are tied together to build a larger one. This can enhance reliability through physical partitioning of the resource pool and address communication latency issues by binding clients to the nearest data centre [9]. Furthermore, federated Clouds are an interesting alternative for those companies who are reluctant to move their data out of house to a service provider due to security and confidentiality concerns. By operating on

geographically distributed data centres, companies could still benefit from the advantages of Cloud Computing by running smaller Clouds in-house, and federating them into a larger Cloud [10].

A Cloud Federation allows final users to access transparently a set of resources and services, distributed among several independent CSP [10]. Rak et al. [10] identify the following actors as the key players in this scenario:

- Final Users: common users which access the Cloud and uses the Cloud services.
- Service Providers: acquire resources and services from the Cloud in a transparent way, and offer them to Final Users.
- Service Developers: develop applications using the Cloud's resources. Sometimes they also use services developed by other parties.
- Cloud Service Providers: Offer Cloud resources and services.

While users focus on optimising the performance of a single application or workflow, such as application throughput and user perceived response time, Cloud providers aim to obtain the best system throughput, use resources efficiently, or consume less energy. Efficient brokering policies will try to satisfy the user requirements and Clouds' global performance at the same time [11].

Thereby, Cloud federation introduces new avenues of research into brokering policies such as those techniques based on ensuring the required QoS level or those aiming at optimising the energy efficiency [11]. The goals of brokering methods and policies in federated Clouds can be found in different domains. Some examples are listed as follows [10]:

- Cost-effectiveness: federated Clouds provide a larger amount of resources, which may help improve cost-effectiveness, e.g. time to completion, increasing the system throughput or optimising resource utilisation.
- Acceleration: federated Clouds can be used as accelerators to reduce application time-to-completion by using Cloud resources to exploit an additional level of parallelism by offloading appropriate tasks to other Cloud resources.
- Conservation: federated Clouds can be used to conserve allocations, within the appropriate runtime and budget constraints.
- Resilience: federated Clouds can be used to handle unexpected situations such as unanticipated downtime, inadequate allocations, or failures of working nodes. Additional Cloud resources can be requested to ease the impact of the unexpected situations and meet user objectives.
- Energy efficiency: federated Clouds can facilitate optimising the energy efficiency of Clouds as multiple objectives can be combined as needed. An example is combining an acceleration objective with a resilience objective.

By providing security services from within the Cloud provider infrastructure, enterprises are able to deploy security policies and rules between each virtual machine or between virtual machine centres. A feature of the Cloud provider infrastructure is that enterprises can maintain corporate security policies and the data collected about them with the virtual machines. This allows them to enforce security services in the enterprise and the Cloud provider consistently.

2.3. Cyber attacks in federated Clouds

Each interface can present specific vulnerabilities that can be exploited by malicious entities, e.g. users, service instances, and CSPs, to perform cyber attacks. The interface between a service instance and an user can be considered as a client-to-server interface, that is vulnerable to all types of attacks that are possible in common client-server architectures, including SQL injection, buffer overflow, privilege escalation, SSL certificate spoofing, phishing attacks, and flooding attacks [12].

The interface between a service instance and a CSP is vulnerable to all attacks that a service instance can run against its hosting Cloud systems, such as distributed denial of service (DDoS) attacks, and Cloud malware injections. In the same way, a malicious CSP of the Cloud Federation may perform several attacks towards service instances running on it. Previous work of ours MacDermott et al. (2014) [13] has highlighted this possibility, conveying how this type of attack could affect interdependent services and CSPs.

DDoS is a serious and growing problem for corporate and government services doing business on the Internet [14]. Targets for DDoS attacks include the computational resources, the memory buffers, the application processing logic, the communications bandwidth, and the network protocol, whereas their effects on the target system are the denial or degradation of provided services [10]. Resource management to prevent DDoS attacks is receiving attention, as the Infrastructure as a Service (IaaS) architecture, effectively 'supports' the attacker. When the Cloud system observes the high workload on the flooded service, it is likely the Cloud federation will start providing more computational power in order to cope with it.

Resource management also has a very important security function, which is to prevent the potential for DDoS attacks. For example, if resource management is not in place, a compromised virtual machine could allow an attacker to starve all of the other virtual machines within that Cloud of their needed resources. By using resource management, a compromised virtual machine can only affect itself and none of the other virtual machines within the Cloud [15]. If the Cloud system notices the lack of availability, it could move the affected service instances to other servers of the Cloud federation. This results in additional workload for such servers, and thus the flooding attack can propagate and spread throughout the whole Cloud federation [11].

Unavailability of services due to Cloud outages can cause monetary loss to Cloud providers and operational loss to Cloud users. Hosting infrastructure services, and storing sensitive data in the Cloud environment brings with it security and resilience requirements that existing Cloud services are not well placed to

address. IDS mechanisms require an extensive use of hardware, especially CPU and memory, and may cause unintentional resource exhaustion or a bottleneck.

2.4. Intrusion Detection and Intrusion Prevention

An Intrusion Detection System (IDS) can be defined as a function that maps the data input into a normal or an attack event either by means of absence of an alert (0) or by the presence of an alert (1) respectively and is given by:

$$\mathrm{IDS} : \mathrm{X} \rightarrow \{0, 1\}.$$

To detect attacks in the incoming traffic, the IDSs are typically parameterised by a threshold T. The IDS uses a theoretical basis for deciding the thresholds for analysing the network traffic to detect intrusions. Changing this threshold allows the change in performance of the IDS. If the threshold is very low, then the IDS tends to be very aggressive in detecting the traffic for intrusions. However, there is a potentially greater chance for the detections to be irrelevant which result in a large number of false alarms. A large threshold on the other hand will have an opposite effect; being a bit conservative in detecting attacks. However, potential attacks may get overlooked by this method [16].

An intrusion prevention system (IPS) operates the process of performing intrusion detection and attempting to prevent detected possible incidents. The IPS is a device or software application that has all the capabilities of an IDS and can also attempt to stop certain incidents. IPSs provide security at all system levels, from the Operating System kernel to network data packets. IPSs also have the ability to prevent known intrusion signatures, besides the unknown attacks originating from the database of generic attack behaviours.

IDSs typically perform extensive logging of data that is related to detected events. This data can be used to confirm the validity of alerts, investigate incidents, and correlate events between the IDS and other logging sources [17]. Discriminating IDSs based on their data sources can be classified as host based and network based. Host based IDSs provide local intrusion detection and support by monitoring user behaviour over an application layer protocol, such as the client-server protocol. Network based IDSs provide global intrusion detection, where they provide level monitoring of traffic flowing through the network and detect intrusions based on the nodes behaviour over the network.

Regardless of whether they operate at the network, host or application level, both IDSs and IPSs use one of two detection methods; anomaly based or signature based. Anomaly based IDSs detect abnormal patterns that deviate from what is considered to be normal behaviour [18]. Anomaly detection does not require prior knowledge of intrusion and can detect new intrusions. However, the drawback is that they may not be able to describe what an attack is and may have a high false positive rate. Signature based IDSs use known patterns of unauthorised behaviour to predict and detect subsequent similar attacks [19]. Such systems can accurately and efficiently detect instances of known attacks. However, they lack the ability to detect zero day attacks. Signature databases must constantly be updated, and IDSs must be able to compare and match activities against large collections of attack signatures.

Weaknesses with present-day IDS techniques is that they do not take into consideration the threat exposures in the network while detecting intrusions, resulting in obtaining alerts for all types of events, many or most of which may not be relevant to the operating environment [20]. In the context of dynamic network environments, this approach may lead to a huge number of unnecessary alerts. Depending on the frequency of changes to the network environment, this may in turn affect the efficacy of the IDS itself.

Additionally, current IDSs are not devised to handle dynamic network environments, as they predominantly use a predefined set of signatures and anomaly detection thresholds to detect intrusions. They can also be ignorant of any changes to the operating environment that may eliminate or introduce vulnerabilities and threat exposures. Due to this weakness, it may miss critical attacks and detect intrusions that are not relevant due to changes to the environment. Hence, the threat-awareness capability provides a key opportunity for an IDS to improve its detection rate [20].

2.5. Contending with Big Data

As we continue to contend with the vast amounts of data and logs generated, by both the monitoring systems and the Cloud user (be it critical infrastructure vendor or corporation), securing this data is imperative. Increasingly, research has been focusing on storing this information for future analysis or Forensic studies.

The growing research area 'Big Data' and its associated data and predictive analytics studies have shown Big Data to have similar issues to that of Cloud security, when it comes to storing their data [21]. The common questions remain, what do we do with this data? How long do we store it for? And how can we ensure it maintains its integrity?

Historical processes generate large datasets [24], but there needs to be sufficient procedures in place for performing data analytics and advice on handling these datasets. Previous work of ours MacDermott et al. (2013) [22] looked into how the Cloud environment could facilitate the storing and study of historical data from critical infrastructure control systems, but the aforementioned issues regarding dataset size and the longevity still remain.

3. Related Work

Since Cloud Computing supports a distributed service oriented paradigm, multi-domain and multi-users administrative infrastructure, it is more prone to security threats and vulnerabilities, such as data breaches, data loss, service hijacking, DDoS attacks, and malicious insiders to name a few [41]. In the Cloud environment, where massive amounts of data are generated due to high network access rates, an IDS must be robust against noise data and false positives. Since Cloud infrastructure have enormous network traffic, traditional IDSs

are not efficient to handle such a large data flow. Due to the large data sets, classification techniques require a huge amount of memory and CPU usage. Much of the proposed academic research on IDSs in the Cloud environment has focused on providing security mechanisms for specific security problems rather than trying to protect the Cloud as a whole.

Hamad and Al-Hoby [23] propose the Cloud Intrusion Detection Service (CIDS), which can be deployed by Cloud providers to enable clients to subscribe with the IDS in a service-based manner. It is a re-engineered version of Snort, which is an open-source network IDS/IPS. The model outperforms currently used solutions for service-based IDS but at the same time provides minimal overhead to the case of traditional IDS deployment for single network protection.

Montes et al. [24] implemented GMonE, a Cloud monitoring tool which capable of being adapted to different kinds of resources, services and monitoring parameters. GMonE performs the monitoring of each element by means of GMonEMon, which abstracts the type of resource (virtual or physical). GMonEMon service monitors the required parameters and then it communicates automatically with the monitoring manager (GMonEDB) to send it the monitored data. This communication is done through a standard Java Remote Method Invocation (RMI) process.

The work of Calheiros et al. [25] conveys an InterCloud project, through the use of agents called Cloud Coordinators and allows for an increase in performance, reliability, and scalability of elastic applications. The architecture proposed for the Cloud Coordinator could be applied to the intrusion detection domain, whereby it has to be present on each data center that wants to interact with InterCloud parties. The Cloud Coordinator is also used by users and brokers that want to acquire resources via InterCloud and do not own resources to negotiate in the market.

The work of Chen et al. [26] aims to develop a new collaborative system to integrate a Unified Threat Management (UTM) via the Collaborative Network Security Management System (CNSMS). Such a distributed security overlay network coordinated with a centralised security center leverages a peer-to-peer communication protocol used in the UTMs collaborative module and connects them virtually to exchange network events and security rules. The CNSMS also has a huge output from operation practice, e.g., traffic data collected by multiple sources from different vantage points, operating reports and security events generated from different collaborative UTMs, etc. There is a vast amount of data generated which is not easy to analyse in real-time, but they also keep it archived for forensic analysis.

Dhage et al. [27] propose an architecture in which mini IDS instances are deployed between each user of the Cloud and the CSP. As a result, the load on each IDS instance will be less than that on a single IDS and for this reason, the small IDS instance will be able to do its work in a more efficient way. For example, the number of packets dropped will be less due to the lesser load which single IDS instance will have. By proposing a model in which each instance of IDS has to monitor only a single user, an effort has been made to create a coordinated design, which will be able to gather appropriate information about the user, thus enabling it to classify intrusions in a better way.

Lee [28] proposes a multi-level IDS and log management method based on consumer behaviour for applying IDS effectively to the Cloud system. They assign a risk level to user behaviour based on analysis over a period of time. By applying differentiated levels of security strength to users based on the degree of anomaly increases the effective usage of resources. Their method proposes the classification of generated logs by anomaly level, so that the system administrator analyses logs of the most suspected users first.

Lo et al. [29] present a cooperative intrusion detection system framework for Cloud Computing networks. They deploy an IDS in each Cloud region, and each entity cooperates with each other through the exchange of alerts to reduce the impact of DDoS attacks. A Snort based IDS is implemented and the three main modules are plugged into the system: block, communicate, defence. A cooperative agent is used to receive alerts from other IDSs, and they are analysed using a majority vote in order to determine the accuracy of results. If deemed a legitimate alert, the blocking rule is implemented. By cooperative operation among these agents, early detection and prevention technique is implemented.

Our analysis has shown that no unified model or unified detection and prevention approaches are established for detecting intrusions in the Cloud environment, nor is there a globally accepted metric or standard to evaluate against. It is clear that an IDS alone cannot protect the Cloud environment from attack. If an IDS is deployed in each Cloud Computing region, but without any cooperation and communication, it may easily suffer from single point of failure attack. Obviously, the abilities of intrusion detection and response are decreased significantly. Thus, the Cloud environment could not support services continually. Based on this concept, intrusion detection services deployed in each Cloud region collaborating is advised. These attributes will cooperate with each other to offer holistic security to those CSPs present, and add to the defence in depth.

4. Security as a Service

Cloud federation, present CSPs will benefit significantly if there is a comprehensive IDS that evolves based on their requirements. The security of applications and services provided in the Cloud, against cyber attacks, is hard to achieve for the complexity, heterogeneity, and dynamic nature of such systems [12]. Distributed collaboration among heterogeneous components within and across independent domains has been indicated in recent literature. The cooperation of threat knowledge, both known attacks and unknown threats; among CSP peers within the enterprise network or with other CSPs will contribute to better incident detection and prevention [13].

This enhances Cloud security and provides faster and more effective incident response. Information sharing in this approach is automated which we conceive to be an important aspect of our approach. Collaboration among CSPs in the federated Cloud could offer holistic security to those providers

in this agreement. Based on a distributed system, collaboration could be used to trace an attack to the source domain [14]. The collaboration of CSPs could help trace the source of attack, identify location, and limit attack vectors.

Cloud defence strategy needs to be distributed so that it can detect and prevent the attacks that originate within the Cloud itself and from the users using the Cloud technology from different geographic locations through the Internet. As the popularity of the services provided in the Cloud environment grows rapidly, the exploitation of possible vulnerabilities grows at the same pace [30]. The measurements required to obtain a comprehensive view on the status of the Cloud lead to the generation of a vast volume of data coming from multiple distributed locations [13].

Attacks and failures are inevitable; therefore, it is important to develop approaches to understand the Cloud environment under attack. The current lack of collaboration among different components within a CSP, or among different providers, for detection or prevention of attacks is the focus of our work. Our current work focuses on the deployment of such a solution for CSP collaboration: Security as a Service.

A Cloud federation requires that each provider has to share Cloud-related information with the federated Cloud providers. This sharing of knowledge in our approach would involve security information about malicious activities, new signatures, and suspicious IP addresses. Our Security as a Service entity would be present in each CSPs domain, and is composed of the following entities: the Cloud Broker, the Command and Control server (C2), the Super Node (SN), and the Monitoring Nodes (MN). A CSP is represented as a domain, and comprises a number of Super Nodes and a C2. A C2 manages its domain, communicates with C2s in other CSP domains, and coordinates a response to an attack.

The Cloud Broker is queried when a decision needs to be made. Rather than communication occurring between the C2s when suspect actions have been observed, the querying C2 would firstly prompt the Broker to check if the actions are legitimate or not. This would keep communication and network overheads down, as there would be an increase in network latency if there were queries every time something suspect was observed. For this reason, we have inferred the hierarchy that we have in our approach.

Figure 1 visualises the levels of communication occurring between each entity in our solution:

Monitoring Nodes

Monitoring nodes deal with issues on a local level and communicate with their neighbouring nodes regarding systems states and signatures. Monitoring nodes contain a black list determined by the Broker, and a local grey list, which contains ambiguous observations.

Monitoring nodes trigger a pre-alarm when a pre-defined threshold is violated. Specifically, a pre-alarm is sent when the observed value is compared with a global threshold, such as using CUSUM for traffic volume dynamics. CUSUM is a widely used anomaly detection algorithm that has its foundations in change point detection. In particular, an alarm is signalled when the accumulated volume of measurements are above some traffic threshold exceeds an aggregate volume threshold. The CUSUM algorithm considers the excess volume above the normal volume, hence accounts for the intensity of the violations.

When a pre-alarm is sent, monitoring nodes add it to their local grey list. Let a monitored value on the monitoring node *i* at time *t* be xi (t), i ∈ [1, n], where n is the number of monitors involved in the monitoring task, and the global threshold be *T*,

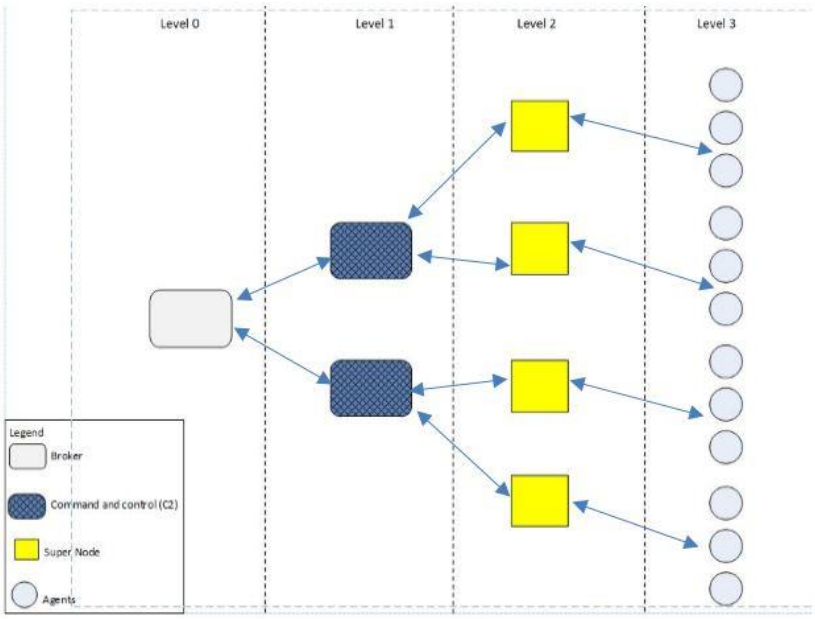

Fig. 1. Levels of communication

it can be considered the state at t to be abnormal and triggers a state alert if $\sum_{i=1}^{n} x_i\ (t) > T$, which we refer to as a global violation [31].

T is decomposed into a set of local thresholds T_i, for each monitor *i* such that $\sum_{i=1}^{n} T_i \leq T$. As a result, as long as $x_i(t) \leq T_i, \forall i \in [1, n]$, i.e. the monitored value is lower or equal to its local threshold, the global threshold cannot be exceeded because $\sum_{i=1}^{n} x_i(t) \leq \sum_{i=1}^{n} T_i \leq T$. In this case, monitors do not need to report their local values to the super node.

When $x_i(t) > T_i$ on monitor *i*, it is possible that $\sum_{i=1}^{n} x_i(t) > T$.

Hence, monitor *i* sends a message to the super node to report a local violation with the value $x_i(t)$.

Super nodes

A Super Node has a parent/child relationship with a monitoring node under its management. The Super Node effectively communicates upstream with the C2 to query any suspicious actions. The hierarchy of communication means network latency is low, and communication occurs only when essential, or when thresholds are violated. The Super Node, based on the amount of monitoring nodes in its subset, observes the generated alarms, these alarms are counted and when the pre-alarm count is more than or equal to the threshold based on the amount on monitoring nodes, a belief is formed that there is an attack. The Super Node then sends this belief to the C2, who queries the Broker.

Command and Control server (C2)

The command and control server (C2) is effectively a domain management node. When a threat is detected in its domain, a belief is formed that an attack is underway. The C2 queries the Broker about the generated belief, to see if it is legitimate or not. C2s possess black lists comprised of attack signatures, and local grey lists provided by the SN and MN which contains ambiguous observations.

Broker

Currently, Cloud Brokers offer tools to manage applications across multiple Cloud providers. In the future, Cloud Brokers will offer services based on their knowledge of the Cloud providers infrastructure [32]. We could use this knowledge to offer Cloud Security as a Service, where the Broker has the knowledge base of Cloud attacks and behavioural profiles to identify threshold violations. The Broker is the security provider. This is propagated to the C2s present in each CSP domain.

The Broker invokes a global poll procedure when a decision cannot be made. He queries the C2s in adjacent domains, and asks them to generate their own beliefs. They check their local grey lists to see if they have encountered the suspect actions previously. Their grey list is a function mapping signatures to beliefs. Each C2 generates their own belief, and the Broker uses Dempster-Shafer to fuse the different beliefs and to create one decision. This in turn can improve resilience to attack. The predefined black lists are of attack signatures, and the monitoring nodes can analyse anomalous actions and threshold violations.

5. Dempster-Shafer theory of evidence

For collaborative detection, we use the Dempster-Shafer (DS) theory of evidence. DS theory is a probabilistic approach, which implements belief functions which are based on degrees of belief or trust. Probability values are assigned to sets of possibilities rather than single events [24]. DS was first introduced as a mathematical framework for the representation of uncertainty. The main advantage of this algorithm is that no

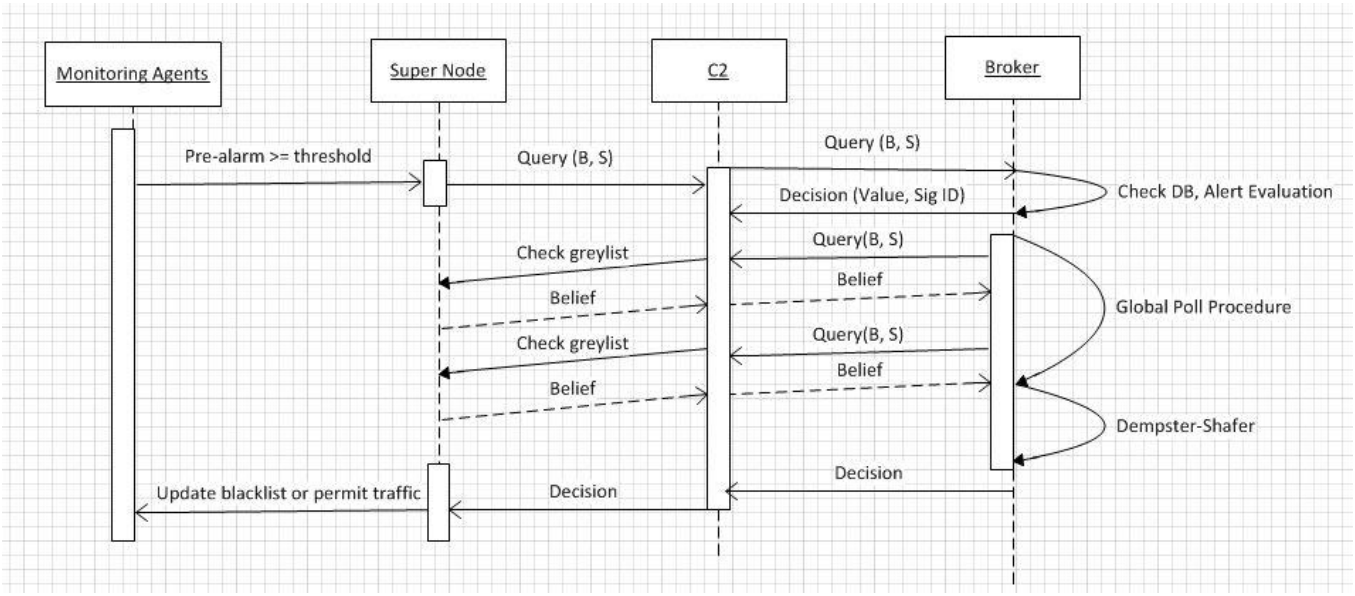

Fig. 2. Collaborative decision process activity diagram

priori knowledge of the system is required, thus making it suitable for anomaly detection of previously unseen information [33]. Collaborative intrusion detection has been considered in several contributions where data provided by heterogeneous intrusion detection monitors is fused.

Our intrusion detection algorithms in our solution are of two types: local detection algorithm and a fusion algorithm. The latter focuses on outputs provided by the local algorithm, thus forming a distributed collaborative intrusion detection method. DS executes as a main fusion node, an entity with the role to collect and fuse the information provided by the monitors, taking the final decision regarding a possible attack. An advantage of DS is its usefulness in combining data sent by different observers.

In the decision making process, the uncertainty existing in the network often leads to the failure of intrusion detection or low detection rate. The DS theory of evidence in data fusion has solved the problem of how to analyse the uncertainty in a quantitative way. Figure 2 illustrates the actions taken in our collaborative decision process, where a C2 is invoked and queries the Broker regarding the suspect behaviour.

Basic concepts of Dempster-Shafer [33] include:

Definition 1 – The frame of discernment:
A complete set describing all of the sets in the hypothesis space. Generally, the frame is denoted as θ. The elements in the frame must be mutually exclusive. While the number of the elements is n, the space will be 2^n.

Definition 2 – Basic probability assignment:
It is a positive number between 0 and 1. It exists in the form of probability. The value of BPA denotes the degree supporting or refuting evidence, and is denoted as $m(A)$.

Definition 3 – Belief function:
For $2^\theta \in [0,1]$, $\mathrm{Bel}(A) = \sum_{B \subseteq A} m(B)$ describes the general belief supporting the hypothesis, where 2^θ is the hypothesis space.

Definition 4 – Plausibility function:
For $2^\theta \in [0,1]$, $\mathrm{Pl}(A) = 1 - \mathrm{Bel}(A^c) = \sum_{B \cap A = \emptyset}$ describes the belief not refuting the hypothesis.

According to the above concepts, the belief function and plausibility function are related by $\mathrm{Bel}(A) \leq \mathrm{Pl}(A)$.

Then we call $[\mathrm{Bel}(A), \mathrm{Pl}(A)]$ the Belief Range.

Dempster-Shafer combination rule:
DS utilises orthogonal sum to combine the evidences [34]. We define the belief functions, describing the belied in a hypothesis A, as $\mathrm{Bel}_1(A), \mathrm{Bel}_2(A)$; then the belief function after the combination is defined as:

$$\mathrm{Bel}(A) = \mathrm{Bel}_1(A) \oplus \mathrm{Bel}_2(A)$$

The mass function after the combination can be described as:

$$m(A) = K^{-1} \cdot \sum_{A_i \cap B_i = A} m_1(A_i)\, m_2(B_j)$$

Where K is called Orthogonal Coefficient, and it is defined as:

$$K = \sum_{A_i \cap B_i \neq \emptyset} m_1(A_i)\, m_2(B_j)$$

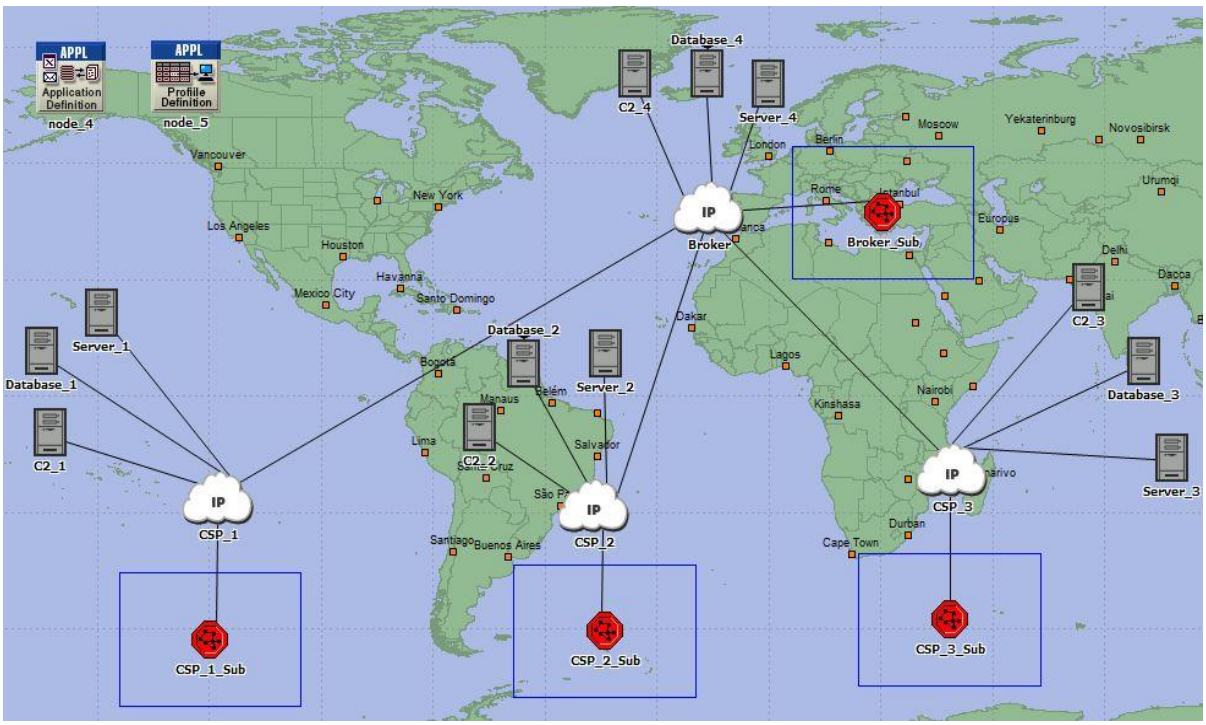

Fig. 3. Overview of attributes in OPNET

DS combines the beliefs expressed by monitors producing a single combined belief that is finally compared with the accumulative sum of the beliefs q. If the combined belief is greater than q, an alarm is raised [35].

The monitors (based on the local detection algorithms) produce a single belief for each focal element:

- b_a: the belief that there is an attack
- b_n: the belief there is not an attack (normal)
- b_{na}: the belief expressing an ambiguity: attack or no attack.

DS's theory of evidence can be regarded as the expansion of Bayesian Inference. The Bayesian inference needs priori knowledge as the foundation of inference. Furthermore, the inference is unable to provide a better way to analyse the "uncertainty" in a quantitative way. DS proposes the concepts: "belief" and "plausibility", which can aid the theory to analyse the "incomplete" or "missing" quantitatively. In this way, the inference can guarantee the accuracy of the decision.

6. Implementation details

In the previous section, the design of the architecture was presented; comprising four tiers; Cloud Broker, Command and Control servers (C2), Super Nodes (SN) and Monitoring Nodes (MN). The main aim of Security as a Service is to provide collaborative intrusion detection in a federated Cloud environment. This is imperative for protecting critical infrastructure services and sensitive data, as their failure or unavailability of such processes has high socioeconomic implications [9].

The system uses a hybrid IDS, and a Cloud Broker to propagate information to the C2 entities in each CSP domain. Monitoring nodes are used to observe the states and processes in the Cloud environment, and update each adjacent domain on any changes or suspicious activities, which they, in turn, would be updated and protected against.

Collaborative security between CSPs in a Cloud federation can offer holistic security to those in this scheme. Information sharing in this approach is automated which we conceive to be an important aspect of our approach. For proof of concept we use a scaled number of entities but for future work we would expand our solution and adapt it to have a self-organising hierarchy. Dividing the system into domains makes the system more scalable. Domain management nodes (C2) may cooperate.

Using OPNET, attributes of our system were implemented. OPNET is a large and powerful software which enables the possibility to simulate heterogeneous networks with various protocols. OPNET consists of a high level user interface, which is constructed from C and C++ source code, and also possesses a library of OPNET specific functions. One specific benefit of using this simulator is that all processes contain code to record performance metrics, which is favourable for observing both local and global statistics in our solution.

Using OPNET, the topology for our solution was implemented as conveyed in Figure 3. This conveys a Cloud federation scenario, where each CSP has an end user in a sub network. The Cloud Broker was depicted as a Cloud entity; in addition, three Cloud Service Providers were added: CSP_1, CSP_2 and CSP_3. Connected to each CSP is a server, database, and a C2. Each Cloud entity contains Cloud network protocols, IP encapsulation, and primary transmitters and receivers. OPNET allows the user to simulate different scenarios and gather data and statistics from the chosen scenario.

The current focus of our work is implementing our collaborative intrusion detection process. We have created an environment that facilitates the Cloud federation, and Cloud service providers present. Hierarchy in a network topology is achieved using subnets, which represent identical constructs in an actual network. Each CSP is connected to a subnetwork, the characteristics of which are illustrated in Figure 4. These allow us to simulate end users of the CSP, and how malicious actions from one could affect the interconnected domains.

The next step is to introduce the roles of MN and SN and finalise the DS algorithm, and global poll procedure in which the Broker queries the C2s. We are simulating from the point where a SN has observed pre alarms and believes there is an attack. Currently the main attributes of our solution have been implemented, and our next aim is to continue to refine the functionality, and test our hypotheses. A simulation study of the effects of DDoS attacks on the performance of the collaborative intrusion detection process and DS theory of evidence is required.

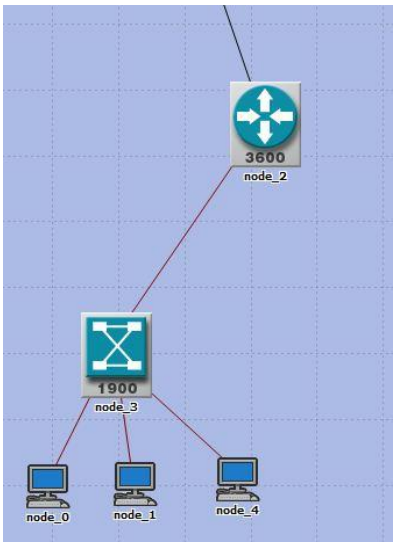

Fig. 4. Subnet of CSP 1

Figure 5 shows the average delay in seconds of the aforementioned topology running. When we introduce our DS algorithm we need to ensure that it is lightweight and resource efficient, and doesn't affect the operations of the Cloud services.

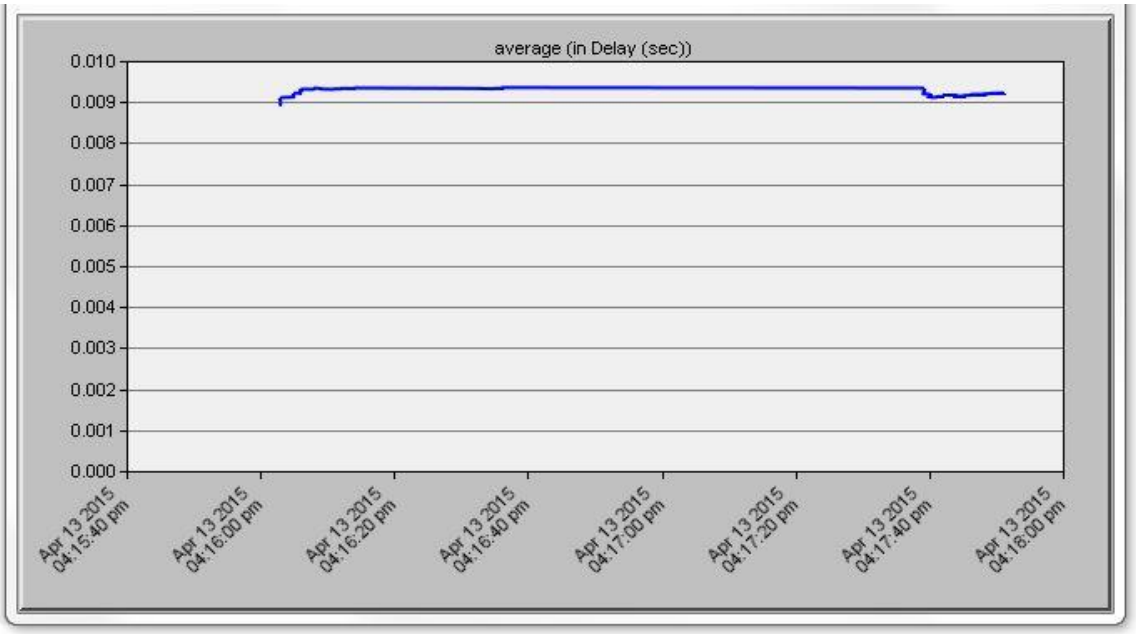

Fig. 5. Average delay in seconds

7. Conclusions

This paper has presented our Security as a Service solution for collaborative intrusion detection in federated Cloud environments. Protecting the federated Cloud against cyber-attacks is a key concern, since there are potential significant economic consequences. For proof of concept we have simulated attributes of our system using OPNET, and are currently implementing the intrusion detection process to prove our hypotheses. Current work in this area uses majority voting when making collaborative decisions, and this is often using binary inputs. Using a Cloud Broker to provide Security as a Service to service providers in a federation we can improve overall resilience to attacks.

Our work involves the use of the Dempster-Shafer theory of evidence which takes in other information, providing more accurate results. Observations from different CSPs are correlated autonomously, in order to determine whether similar behaviour that is indicative of an attack or other issues have been observed in their domains. The integration of the decisions coming from different IDSs has emerged as a technique that could strengthen the final decision. Federated Cloud environments are growing areas in terms of adoption by critical infrastructure vendors, and large corporations, so our Security as a Service facilities this collaborative intrusion detection, and sharing of attack information among these different service providers.

8. References

[1] D. Wallom, M. Turilli, A. Martin, A. Raun, G. Taylor, N. Hargreaves, and A. McMoran, "myTrustedCloud: Trusted Cloud Infrastructure for Security-critical Computation and Data Managment," 2011 IEEE Third Int. Conf. Cloud Comput. Technol. Sci., pp. 247–254, Nov. 2011.

[2] OTE, "Discussion on the Challenges for the Development of a Context for : Secure Cloud computing for Critical infrastructure IT," Greece, 2012.

[3] S. Paudel and M. Tauber, "Security Standards Taxonomy for Cloud Applications in Critical Infrastructure IT," in 8th International Conference for Internet Technology and Secured Transactions (ICITST), 2013, pp. 645 – 646.

[4] M. Sch, R. Bless, F. Pallas, J. Horneber, and P. Smith, "An Architectural Model for Deploying Critical Infrastructure Services in the Cloud," in IEEE Cloud Com 2013, 2013.

[5] M. T. Khorshed, a. B. M. S. Ali, and S. a. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," Futur. Gener. Comput. Syst., vol. 28, no. 6, pp. 833–851, Jun. 2012.

[6] S. Wright, "ETSI NFV ISG," The Internet Engineering Task Force (IETF) IETF 91 Proceedings, 2014. [Online]. Available: https://www.ietf.org/proceedings/91/slides/slides-91-nfvrg-8.pdf. [Accessed: 12-Feb-2015].

[7] K. Hwang, S. Kulkareni, and Y. Hu, "Cloud Security with Virtualized Defense and Reputation-Based Trust Mangement," 2009 Eighth IEEE Int. Conf. Dependable, Auton. Secur. Comput., pp. 717–722, Dec. 2009.

[8] Á. MacDermott, Q. Shi, M. Merabti, and K. Kifayat, "Protecting critical infrastructure services in the cloud environment considerations," Inderscience Int. J. Crit. Infrastructures, vol. 10, no. 3, 2014.

[9] O. Babaoglu, M. Tamburini, and U. Bologna, "Design and Implementation of a P2P Cloud System," in Proceedings of the 27th Annual ACM Symposium on Applied Computing, 2012, pp. 412–417.

[10] M. Rak, M. Ficco, J. Luna, H. Ghani, N. Suri, S. Panica, and D. Petcu, "Security Issues in Cloud Federations," in Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice, 2012, pp. 176–194.

[11] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. Masoud Sadjadi, and M. Parashar, "Cloud federation in a layered service model," J. Comput. Syst. Sci., vol. 78, no. 5, pp. 1330–1344, Sep. 2012.

[12] N. Gruschka and M. Jensen, "Attack Surfaces: A Taxonomy for Attacks on Cloud Services," in 2010 IEEE 3rd International Conference on Cloud Computing, 2010, pp. 276–279.

[13] Á. Macdermott, Q. Shi, M. Merabti, and K. Kifayat, "Security as a Service for a Cloud Federation," in The 15th Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet2014), 2014, pp. 77–82.

[14] N. Kumar, "Study of Intrusion Detection System for DDoS Attacks in Cloud Computing," in 2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN), 2013.

[15] VM Ware Inc., “Securing the Cloud a Review of Cloud Computing, Security Implications and Best Practices,” Tech Republic, Whitepaper, 2003. [Online]. Available: http://www.techrepublic.com/resource-library/whitepapers/securing-the-cloud-a-review-of-cloud-computing-security-implications-and-best-practices-copy1/. [Accessed: 25-Jul-2013].

[16] C. Thomas and B. Narayanaswamy, “Sensor Fusion for Enhancement in Intrusion Detection,” in Sensor Fusion - Foundation and Applications, 2011, pp. 61–76.

[17] F. Sabahi and A. Movaghar, “Intrusion Detection: A Survey,” in 2008 Third International Conference on Systems and Networks Communications, 2008, pp. 23–26.

[18] A. Patel, Q. Qassim, and C. Wills, “A survey of intrusion detection and prevention systems,” Inf. Manag. Comput. Secur., vol. 18, no. 4, pp. 277–290, 2010.

[19] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A Survey,” ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, Jul. 2009.

[20] S. Neelakantan and S. Rao, “A Threat-Aware Hybrid Intrusion – Detection Architecture for Dynamic Network Environments,” CSI J. Comput., vol. 1, no. 3, 2012.

[21] H. Cheng, C. Rong, K. Hwang, W. Wang, and Y. Li, “Secure big data storage and sharing scheme for cloud tenants,” China Commun., vol. 12, no. 6, pp. 106–115, 2015.

[22] Á. MacDermott, Q. Shi, M. Merabti, and K. Kifayat, “Protecting Critical Infrastructure Services in the Cloud Environment,” in Proceedings of the 12th European Conference on Information Warfare and Security, 2013, pp. 336–343.

[23] H. Hamad and M. Al-Hoby, “Managing Intrusion Detection as a Service in Cloud Networks,” Int. J. Comput. Appl., vol. 41, no. 1, pp. 35–40, Mar. 2012.

[24] J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, and G. Antoniu, “GMonE: A complete approach to cloud monitoring,” Futur. Gener. Comput. Syst., vol. 29, no. 8, pp. 2026–2040, 2013.

[25] R. N. Calheiros, A. N. Toosi, C. Vecchiola, and R. Buyya, “A coordinator for scaling elastic applications across multiple clouds,” Futur. Gener. Comput. Syst., vol. 28, no. 8, pp. 1350–1362, 2012.

[26] Z. Chen, F. Han, J. Cao, X. Jiang, and S. Chen, “Cloud computing-based forensic analysis for collaborative network security management system,” Tsinghua Sci. Technol., vol. 18, no. 1, pp. 40–50, 2013.

[27] S. N. Dhage and B. B. Meshram, “Intrusion detection system in cloud computing environment,” Int. J. Cloud Comput., vol. 1, no. 2/3, p. 261, 2012.

[28] J. Lee, M. Park, and J. Eom, “Multi-level Intrusion Detection System and log management in Cloud Computing,” 2011 13th Int. Conf. Adv. Commun. Technol., no. 1, pp. 552–555, 2011.

[29] C.-C. Lo, C.-C. Huang, and J. Ku, “A Cooperative Intrusion Detection System Framework for Cloud Computing Networks,” in 2010 39th International Conference on Parallel Processing Workshops, 2010, pp. 280–284.

[30] S. Taghavi Zargar, H. Takabi, and J. Joshi, “DCDIDP: A Distributed, Collaborative, and Data-driven Intrusion Detection and Prevention Framework for Cloud Computing Environments,” Proc. 7th Int. Conf. Collab. Comput. Networking, Appl. Work., pp. 332–341, 2011.

[31] S. Meng, A. K. Iyengar, I. M. Rouvellou, L. Liu, K. Lee, B. Palanisamy, and Y. Tang, “Reliable State Monitoring in Cloud Datacenters,” in 2012 IEEE Fifth International Conference on Cloud Computing, 2012, pp. 951–958.

[32] M. Mechtri, D. Zeghlache, E. Zekri, and I. J. Marshall, “Inter and intra Cloud Networking Gateway as a service,” in 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet), 2013, pp. 156–163.

[33] Q. Chen and U. Aickelin, “Anomaly Detection Using the Dempster-Shafer Method,” in DMIN, 2006, pp. 232–240.

[34] W. H. Jianhua Li and Q. Gao, “Intrusion Detection Engine Based on Dempster-Shafer’s Theory of Evidence,” in 2006 International Conference on Communications, Circuits and Systems Proceedings, 2006, vol. 2, no. 2003, pp. 1627–1631.

[35] A. G. Fragkiadakis, V. a. Siris, N. E. Petroulakis, and A. P. Traganitis, “Anomaly-based intrusion detection of jamming attacks, local versus collaborative detection,” J. Wirel. Commun. Mob. Comput., vol. 15, no. 2, pp. 276–294, Jan. 2013.