Taxonomy and Proposed Architecture of Intrusion Detection and Prevention Systems for Cloud Computing

Ahmed Patel[1,2,*], Mona Taghavi[3], Kaveh Bakhtiyari[4], and Joaquim Celestino Júnior[5]

[1,3,4] School of Computer Science, Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor D.E., Malaysia
[2] Visiting Professor, School of Computing and Information Systems
Faculty of Science, Engineering and Computing
Kingston University, Kingston upon Thames KT1 2EE, United Kingdom
[5] Vieira Computer Networks and Security Laboratory (LARCES),
State University of Ceará (UECE), Fortaleza, Ceará, Brazil

Abstract. The distributed and open structure of cloud computing and services becomes an attractive target for potential cyber-attacks by intruders. The traditional Intrusion Detection and Prevention Systems (IDPS) are deemed largely inefficient to be deployed in cloud computing environments due to their openness, dynamicity and virtualization in offered services. This paper surveys and explores the possible solutions to detect and prevent intrusions in cloud computing systems by providing a comprehensive taxonomy of existing IDPS. It discusses the key features of IDPS that are challenging and crucial for choosing the right security measures for designing an IDPS. The paper further reviews the current state of the art of developed IDPSs for cloud computing which uses advanced techniques in overcoming the challenges imposed by cloud computing requirements for more resilient, effective and efficient IDPSs, abbreviated as CIPDS.

Keywords: Intrusion detection, intrusion prevention, cloud computing, taxonomy, architecture, autonomic techniques.

1 Introduction

Cloud computing is defined as a geeky term for the internet that allows linking all cloud services together to access data anywhere and anytime through a myriad of portable devices. It involves multi-mesh distributed and service oriented paradigms, multi-tenancies, multi-domains and multi-user autonomous administrative infrastructures which are far more vulnerable and prone to security risks than previously thought of. Cloud computing can also be exposed to a multitude of system and non-system threats including threats to the integrity, confidentiality and availability of its resources, data and the virtualized infrastructure which can be used as a launching pad for new attacks [1]. During 2011, a hacker used Amazon's Elastic Computer Cloud service to attack Sony's online entertainment systems by registering

* Corresponding author: whinchat2010@gmail.com

Y. Xiang et al. (Eds.): CSS 2012, LNCS 7672, pp. 441–458, 2012.

and opening an Amazon account and using it anonymously [2]. Cloud services are as cheap and convenient for hackers as they are for service customers. This malicious incidental attack on Sony compromised more than 100 million customer accounts, the largest data breach in the U.S. Some high-profile cases prove how dangerous *cloud living* can be!

In this struggle to secure the systems in cloud computing, IDPS can prove to be an invaluable tool, where its goal is to perform early detection of malicious activity and possibly prevent more serious damage to the protected systems [3]. By using IDPS, one can potentially identify an attack and notify the appropriate personnel immediately or prevent it from succeeding, so that the threat can be contained. This research amalgamates different ways of developing IDPS specifically targeting distributed systems and cloud computing environments by proposing an architecture using advanced techniques to overcome challenges specific to such environments.

2 Intrusion Detection and Prevention Systems Taxonomy

Attacks that come from external origins are called outsider attacks. Insider attacks involve unauthorized internal users attempting to gain and misuse non-authorized access privileges. Intrusion detection is the process of monitoring computers or networks for unauthorized entry, activity or file modification. Attacks mostly occur in distinctive groups called incidents. Although many incidents are malicious in nature, many others are not; for example, a person might mistype the address of a computer and accidentally attempt to connect to a different system without authorization. Fig. 1 provides a high level taxonomy of IDPSs.

2.1 Functional Layer

As Fig.1 (excluding the dashed boxes) shows, IDPSs perform four essential security functions in real-time: they monitor, detect, analyze and respond to unauthorized activities as presented in the functional layer.

IDPSs identify attacks while the system or network is being monitored for intrusions and can immediately flag any deviations and provide proper prevention. The detection process typically outstrips the progress of attacks but cannot handle encrypted packets without more intensive processing. However, IDPSs are also run for deeper off-line analysis inspecting and sieving through historical data to identify past intrusions to update prevention profiles for subsequent use. By contrast, in a non-real-time detection audit, the data is processed with delay, which has high capabilities to provide evidence of data forensic but cannot provide real time response to prevent or mitigate damages.

Audit data can be collected from a single source in a centralized fashion, or in a distributed approach from several different locations. The drawback of a distributed approach is that the data flow between the host monitors and the director agent may generate significantly high network traffic overheads, while for the central approach

an intruder can modify or disable the programs running on a system, making the IDPS useless or unreliable.

The data collected in the monitored environment for analysis can be of three types:

1. *Network-based (NIDPS)* monitors network traffic for particular devices or network segments and analyze the network and application protocol activity to identify suspicious activity. Its strategic position allows for quick response, but it does not have a full picture of the network topology between the other NIDPSs and the hosts, so they may unable to determine a given packet received by a host.
2. *Host-based (HIDPS)* monitors the dynamic behavior and the state of a computer system. Much as an NIDPS will dynamically inspect network packets, an HIDPS might detect which program accesses what resources. There is also a complementary approach that combines NIDPS and HIDPS to provide greater flexibility in deployment. Although it has a very limited view of the network, it is easy to deploy and see low-level local activities such as file accesses and changes to file permissions.
3. *Application-based (AIDPS)* concentrates on the events which occur in some specific applications by analyzing their log files or measuring their performance. The data sources of running applications are its input. This approach is useful when the data of the user-side is only available and the service provider is not willing to impart any information.

There are three models for threat detection:

1. Misuse detection uses known patterns of unauthorized behavior, called signatures, to predict and detect subsequent similar attempts. It generates a very low false positive alarm rate, but it has severe limitation in detection of unknown attacks (called *zero-days*).
2. Anomaly detection is designed to discover abnormal behavior patterns. IDPS establishes a baseline of normal usage patterns, and whatever deviates from this is flagged as possible intrusions. Any incident that occurs on frequently greater or less than two standard deviations from the statistical norm is considered to be an anomaly [4]. A further refinement is for the threshold value to be applied according to the Euclidean distance for *incidents* and the standard deviation value to detect the *anomalies*. The lower threshold value shows that the incidents are closely related to the *normal activities* and a higher threshold value detects more *severe anomalies*. Anomaly techniques use fewer rules compared to the signature based techniques. These techniques increase the detection accuracy rates with greater effectiveness. In turn, they have higher false positive alarm rates since it is too difficult to discover the boundaries between abnormal and normal behavior. There are various categories of anomaly detection proposed, but the three most commonly used ones are [5]:

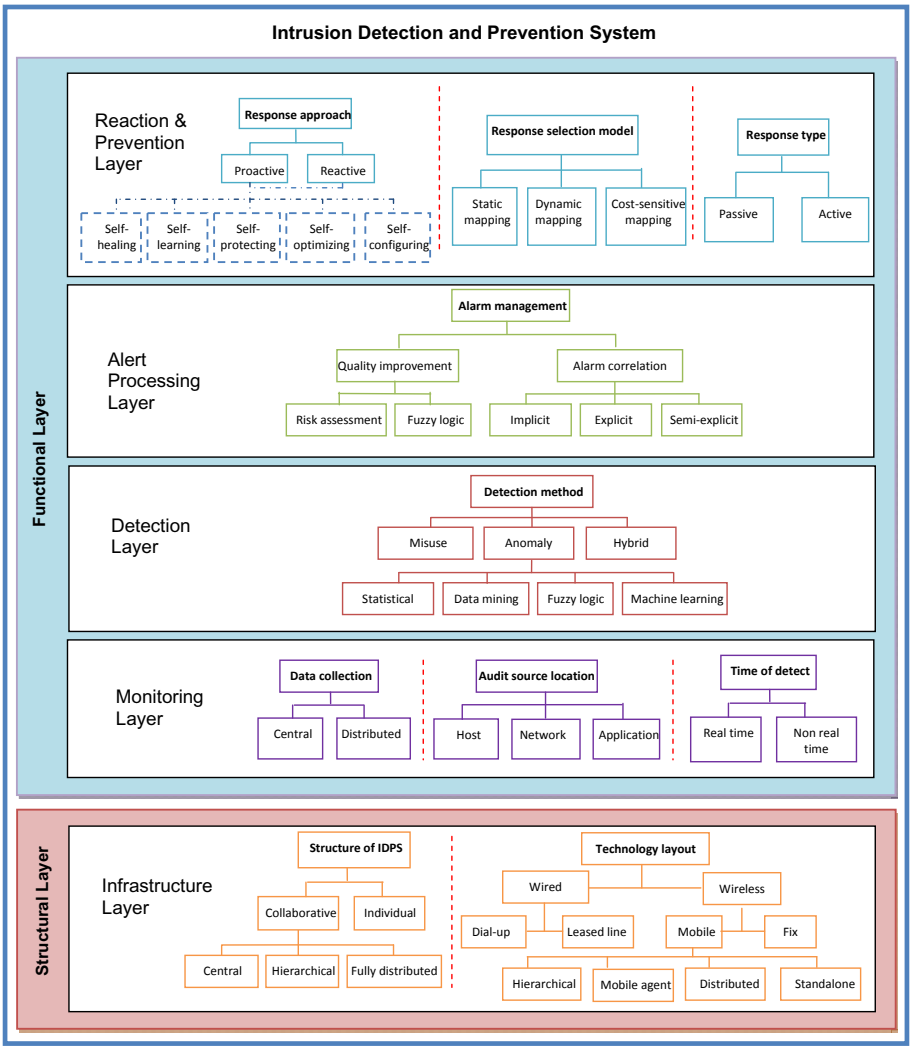

Fig. 1. A layered-taxonomy of CIDPS

- Statistical where the system monitors the activity of subjects (such as CPU usage or the number of TCP connections) in terms of statistical distribution and creates the profiles of their behaviors. Thus, they make two profiles: one is made during the training phase and the other is the current profile during the detection. An anomaly is recognized if there is a difference between these two profiles.
- Machine learning where the system adaptively learns to improve its performance over time. It tends to focus on making a system which can optimize its performance during a loop cycle and can change its execution strategy according to feedback information. The most frequently used

techniques are System call-based sequence analysis, Bayesian network and the Markov model.
- Data mining where the system can help to improve the process of intrusion detection by uncovering patterns, associations, anomalies, changes, important events and structures in the data patterns. Classification, clustering outlier detection and association rule discovery are data mining techniques used in IDPS.

3. Hybrid approach enhances the capabilities and performance of the available IDPS by combining the two methods of misuse and anomaly. The main idea is that misuse detects known attacks while anomaly detects unknown attacks.

Alarm management can be classified into two methods [6]:

1. Alert (alarm) quality improvement: this method improves the alert quality by using additional information, such as vulnerability reports or alert context. Although this method is simple to implement and adoptable in most of the current alert correlation systems, but using it individually, it becomes inefficient to handle false positive alarms. Typically, it works within the context and in cognizance of:
   - Risk assessment: this approach uses risk analysis and risk assessment to generate vulnerability reports and match them with correlated alerts. Lippmann et al. suggested prioritizing alerts according to the vulnerabilities of the victim in a way that correctly identifies intrusions which are given lower priority or discarded if that specific victim is not vulnerable to that attack [7].
   - Fuzzy logic: it analyzes the alarms and vulnerabilities by a fuzzy aspect. Defining the level of severity for each individual incident is a responsibility of fuzzy processing to give a proper response. Non-fuzzy IDPS sets a fixed platter line of threshold which is not a suitable solution, but the fuzzy IDPS auto-set the threshold value for anomaly detection.
2. Alarm correlation: this method reconstructs the high-level incidents from low-level alerts. For some attacks, IDPS generates many alarms. Assume that a set of alerts are trigged, knowing this only without any additional background knowledge, one cannot make certain whether these are single coordinated attacks, or independent attacks that happen to be interleaved. If it is a single attack, then alerts would have to be gathered as a single incident. But, in the case of multiple attacks, the alerts should be divided up to multiple incidents, namely, one incident per attack. Grouping alerts that constitute a single attack into a single meta-alert is aggregation. The task of clustering alerts into incidents is called correlation which tries to explain *events*. Auto Correlation Function can also be used to determine the repeated patterns of incidents to generate proper alarms for the possible series of attacks rather than generating multiple alarms. The main issue of this method is that most of the proposed algorithms in the current literature on correlation match the attack information provided by misuse detectors [8]. Alarm correlation can be performed in three ways:

- Implicit: it uses data-mining techniques to analyze, aggregate and cluster large alert datasets. However, this method fails to enhance the semantics of the alerts, but it is suitable for analysis of huge numbers of alerts.
- Explicit: this approach relies on language allowing security experts to specify logical and temporal constraints between alert patterns to identify complex attack scenarios.
- Semi-explicit: this approach is an extension of the explicit approach which associates preconditions and postconditions, represented by the first order formulae, with individual attacks or actions. Hence, it assumes that complex intrusion scenarios are likely to involve attacks whose prerequisites correspond to the consequences of some earlier ones. The correlation process receives individual alerts and tries to build alert threads by matching the preconditions of some attacks with the postconditions of some prior ones.

When an IDPS responds actively to an intrusion, it may modify the attacked system state further or, in rare cases, modify the attacker state by removing his/her platform. In some cases, they can instruct the network security devices to reconfigure themselves to block certain types of activities or route them elsewhere. They may reconfigure network firewalls by changing the user access control policy temporarily when an attack occurs. Active response may delay the benign traffic unnecessarily since alarm events are blocked. Passive systems can attempt to terminate the connection before an attack can succeed, for example, by ending an existing TCP session. Passive response exposes the assets to the attacks while the security administrator investigates the alarms.

To respond to an attack two approaches can be considered, the reactive approach delays all responses until the intrusion is detected. This approach fails to provide high protection. For instance, assume that an attacker successfully accesses a database and read critical information. The administrator then receives an alarm regarding a malicious activity. Since the critical information has already been disclosed, a reactive response is not useful in this case. By contrast, a proactive approach prevents a malicious activity before it occurs.

There are three models for selecting a proper response:

1. Static mapping: in this model a generated alert is mapped to a predefined response. The main drawback of this model is that the attacker can predict the response measures.
2. Dynamic mapping: responses to an attack may differ for different targets and several factors affect the ultimate response, such as attack metrics (frequency and severity), system state and network policy. The main problem of this model is that it does not learn anything from the attacks, so the intelligence level remains the same until the next update.
3. Cost-sensitive mapping: this model trades-off intrusion damage and response cost. It has two approaches to assess the risk. Offline risk assessment evaluates all the resources in advance, so the value of each resource is static. Meanwhile, online risk assessment accurately measures intrusion damage in real-time. The only issue is to update the cost factor (risk index) over time.

2.2 Structural Layer

Referring to Fig.1, the infrastructure layer consists of the technology and structure of an IDPS. The technology layout is rarely discussed by the researchers, but given its importance to deploy on a cloud environment, it was investigated through our review.

There are two types of wired connection: dial up through the public switched telephone network; and direct connection through a dedicated or leased line which is an analog compatible point to point connection. In wired networks, the features like traffic behavior and network topology can be employed in detecting of intrusions. They are fast and low cost, but heavily dependent on structure platform and not easy to deploy. A mobile ad-hoc network is a collection of mobile nodes that automatically self-configure without assistance of a central management of infrastructure. It is scalable and offers wide coverage and unlimited access which implicates openness to attacks. The wireless network IDPSs are of different sorts including:

- Stand-alone: IDPSs identify intrusion by running on each node independently.
- Distributed: each node participates in detecting intrusion cooperatively and responds through a central IDPS agent.
- Hierarchical, they are deployed in multi-layered networks divided into clusters in which a cluster-head is responsible for its local nodes.
- Mobile agents: they are able to move through a large network, but with a specific task. Different agents have different functionality.

The structure of an IDPS is based on two types: individual or collaborative. An individual arrangement of an IDPS is achieved by physically integrating it within a firewall. Individual IDPS produces more irrelevant and false alarms, but has the advantage of being easy to deploy. A collaborative IDPS consists of multiple IDPSs over a large network where each one communicates with each other. Each IDPS has two main functional components: element detection and correlation handler. Detection elements consist of several detection components which monitor their own sub-network or host individually and generate low level alerts. Then the correlation handler transforms the low level alerts into a high level report of an attack. The issue is that they are less scalable and may have different outputs from different IDPSs for an attack but they are more efficient to detect and prevent intrusions over the Internet. Collaborative IDPSs can be divided into three categories as follows [5]:

1. Central: each IDPS acts as a detection element where it produces alerts locally. The generated alerts are sent to a central server that plays the role of a correlation handler to analyze them. Through a centralized management control an accurate detection decision can be made based on all the available alerts information. The main drawback of this approach is that the central unit is vitally vulnerable. Any failure in the central server leads to deactivating the whole process of correlation. In addition, the central unit should handle the high volume of data which it receives from the local detection elements in a certain amount of time.
2. Hierarchical: the whole system is divided into several small groups based on similar features such as: geography, administrative control, and similar software

platforms. The IDPSs in the lowest level work as detection elements, while the IDPSs in the higher level are furnished with both a detection element and a correlation handler, and correlate alerts from both their own level and lower level. The correlated alerts are then passed to a higher level for further analysis. This approach is more scalable than the centralized approach, but still suffers from the vulnerability of a central unit. Besides, the higher level nodes have higher level abstraction of the input which limits their detection coverage.

3. Fully distributed: there is no centralized coordinator to process the information, it compromises fully autonomous systems with distributed management control. All participating IDPSs have their own two main function components (detection and correlation handlers) communicating with each other. The advantages of the fully distributed IDPS is that the network entities need not have complete information of the full network topology; thus allowing a more scalable design since there is no central entity responsible for doing all the correlation works; and the local alarm correlation activities is simpler in this structure[9]. Meanwhile, fully distributed approach has its own drawback issues [10]: a) the information of all alerts is not available during the detection decision making, so the accuracy might be reduced; b) the alert information usually has a single feature like an IP address which is too narrow for detecting large scale attacks, but it can also have a combination of features such as port #, packet size, types of IP packet and so on to widen the detection of large scale attacks with higher precision and success. The latter is true when using self-learning mechanisms which update their knowledge base.

The proposed taxonomy encompasses new features to help improve the CIDPS design as well as cloud security to neutralize the attacks. The next section identifies some of the important challenges of CIDPS and the proposed solutions and techniques to overcome these challenges.

3 Challenges Imposed by Cloud Computing on IDPS (CIPDS)

It is very important to identify the challenges which originate from cloud computing phenomena before developing a CIDPS. Clouds are defined as large scale Virtual Machine (VM) based systems which are automatically created, migrated and deleted on demand of a user at runtime. Generally, it is supposed that the middleware manager initially is informed from the changes in the resources, but in cloud computing which involves large scale networks and systems, it is crucial to maintain these changes automatically without human intervention. Due to dynamic essence of the monitored systems, the policies should not be static since the *security requirements of each VM tend to be varied* [11].

The *shared infrastructure and virtualization technology* increases vulnerability on cloud computing. Any flaw in the hypervisors, which allows creating virtual machines and running multiple operating systems, exposes inappropriate access and control to the platform [12]. Additional issues concern *visibility into the inter VM traffic on a virtual host platform*, since the switch is also virtualized. Thus, traditional solutions

for physical monitoring are not able to inspect this network traffic [13]. Besides, the new platforms of virtualization themselves would have vulnerabilities that may lead to a big compromise, therefore, they should be monitored and assessed for configuration errors, patches, malware code insertions, preemptive DDoS attacks, etc.

A very important issue in cloud computing is *data transfer cost* [14]. For example, in Amazon Cloud the data transfer cost is about $100 to $150 per terabyte. Therefore, new research should try to provide data cost effective solutions for IDPS in a cloud environment by reducing the network bandwidth.

Usually each company maintains the security procedures to provide a risk profile. But, *cloud service providers are not willing to provide the security log, audit data and security practices* [15]. Lack of transparency on security management practices such as auditing, security policies, logging, vulnerability and incident response leads to inefficiency of traditional risk management techniques in the absence of costumer awareness [1]. In addition, *tracking data across different platform visibility* and *access policies* of different service providers as well as different software and hardware abstraction layers within one provider is a challenging task [16].

A CIDPS should be *scalable* in order to efficiently handle the massive number of network nodes present in cloud and their communication and computational load. It must scale as nodes are added into a larger growing cloud. The placement of detection and correlation handler also affects the scalability and performance of CIDPS.

The feature of *easy to adapt* IDPS in the cloud context to the extent that it operates effectively and efficiently is very important. A CIDPS should configure itself and be adaptive to configuration changes as computing nodes are dynamically added and removed. Designing a suitable architecture of a collaborative IDPS would determine how the alerts should be processed and shared from individual detection components with maintaining a topological model of cloud computing. This also facilitates monitoring and controlling network components as well as the applications in the host. Design of such a system should be flexible enough to be able to accommodate future requirements, challenges and evolving standards.

4 State of the Art of CIDPS

Most of the current proposed IDPSs which work on cloud operate at each of the infrastructure, platform, and application layers separately, and they mainly support detection and prevention independently from the other layers [17]. For operating CIDPS in the infrastructure layer, Tupakula et al. proposed a model based on a VM monitor, called hypervisor, to protect CIPDS from different types of attacks in this layer (IaaS) [18]. Their model improved the reliability and availability of the system, because the infrastructure can be secured most of the time, and running the services can be reliant on the secure infrastructure. This model has not presented any solution to heal the system in case of infrastructure collapse due to the large number of severe attacks on the system. A VM monitor solution embeds as a software layer to control the physical resources and it allows running multiple operating systems. The VM machine monitors are capable of improving the efficiency of intrusion attack

detection and prevention in CIDPS because they have complete control of the system resources and good visibility of the internal state of the VMs.

Majority of the researchers have overlooked the prevention capability in their proposed systems. Gustavo & Miguel implemented several anomaly-based intrusion detection techniques, and presented an IDS for a reasonably complex Web application designated as SaaS [19]. They found the anomaly-based intrusion detection technique as a promising technique to be used in the application layer. They believe that the intrusion on a system occurs where the application code is running; and they interpret the application intrusion as the most potential attack, which may change or inject the false data into the cloud computing system. But they did not suggest any solution for prevention of the attacks. Machine learning is the other method which has been used to train the system for anomaly detection. Vieira et al. proposed a Grid and Cloud Computing Intrusion Detection System (GCCIDS), which covers attacks by using an audit system through integrating hybrid misuse and anomaly method to detect specific intrusions [20]. The authors used Artificial Neural Network (ANN) to train the system and developed a prototype model by using a middleware called Grid-M. They proved that their system had low processing cost while maintaining satisfactory performance for real-time implementation, since it performed the analysis individually on each node, resulting in lower data exchanges between nodes, thus decreasing the complexity of the system. This solution overcomes the challenge of *data transfer cost* since it performs an audit data analysis individually in each node that reduces data transfers and network bandwidth usage. The drawbacks of GCCIDS are that it can detect only specific intrusions, and lacks the ability of prevention against attacks. Although GCCIDS is proposed for both grid and cloud environments, they are different in terms of their security policies, systems requirements and business models [16]; which compels for a specific IDPS design for cloud and grid networks to be performed separately.

Determining the CIDPS structure is always a confusing task for researchers who develop IDPS for cloud computing due to its heterogeneous nature and virtualization. Xin et al. developed a collaborative IDS with a central management approach which provided fast and accurate detection [21]. In spite of the authors' claim about the system's scalability, it is not scalable since the performance decreases with an increase of data load into the central manager node. In addition, the central manager is the single point of failure which is not appropriate in cloud computing. Dhage et al. proposed an individual IDS structure for each user of cloud computing services. In this structure, there is a single controller to manage the instances of IDSs which employs the knowledge base and ANN technique to match the pattern multiple false login attempts and access right violations [22]. Their proposed structure suffers from the challenges of lack of scalability and sensitivity of central management failure. In contrast with this structure, the system which was developed by Kholidy and Baiardi [23] had no central manager coordinator. Their fully distributed system provided a flexible, robust and elastic solution for cloud computing with P2P network architecture, hybrid detection techniques using network and host based audit data. Although their system is scalable but it is not sufficient for detecting large scale distributed attacks on cloud since it processes limited alert information features and

there is no central correlation handler to amalgamate all the alert information consistently to detect intrusions. They do not provide any solution for prevention.

Providing autonomic computing solutions has recently attracted researchers to design, build and manage CIDPS with minimal human intervention. An autonomic system should be capable of adapting its behavior to suit its context of use through methods of self-management, self-tuning, self-configuration, self-diagnosis, and self-healing [24]. Autonomic approaches are particularly suitable to be used in cloud computing systems, where rapid scalability is required across a pool of resources to support various unpredictable demands, and where the system should automatically adapt to avoid failures in the underlying hardware impacting on the user's experience. Autonomic clouds emerge as a result of applying autonomic computing techniques to cloud computing, resulting into robust, fault tolerant and easy to manage and operate cloud architectures and deployments. An autonomic mechanism for anomaly detection in a cloud computing environment was proposed by Smith et al. [25]. They presented a set of techniques to analyze the collected data automatically. This approach provided a uniform format for data analysis, extracted features for data size reduction. It also learnt how to detect the nodes which have abnormal behavior and act differently from others in an unsupervised mode. They made a prototype to evaluate the performance of their mechanism. The results of their evaluation proved the efficiency of their mechanism to detect faulty nodes with low computation overhead and high accuracy due to the reduced data size and machine learning methods. The major drawback of their system is that it does not perform intrusion prevention; it does only detection.

Using ontology enables characterizing knowledge as a set of concepts and relating within the intrusion detection and prevention domain. Martínez et al. presented a model for malware detection, named as uCLAVS, based on intrusion ontology representation for cloud computing Web services [26]. Their idea refers to a new concept in IDPS as an engine which means a processing core and usually it is a file analysis service host. This provides a multi-engine based file analysis service which sends the system files to the network to be analyzed by multiple engines instead of running complex software on every host to analyze them individually. Their model of integrating multiple concepts, relations and managements methods by using ontology is an interesting solution to integrate autonomous IDPSs with a set of common meanings to achieve a set of common goals. Azmandian et al. used data mining techniques and presented a new method in designing IDS for virtual server environments, which utilizes information available from the virtual machine monitor. Their proposed technique supports high detection accuracy with least false alarms, but it trades-off a lack of program semantics for greater malware resistance and ease of deployment [27]. Using a real-time self-learning ontology could fill this semantic gap.

Some of the researchers utilized the available resources and optimized the response through risk assessment and analysis. Lee et al. proposed a multi-level IDS and log management by applying different levels of security strengths to limit the access rights based on the anomaly level and severity of cloud network users or potential intruders [28]. It means that generated logs by the intruder who has the highest anomaly level or security risk are audited with higher priority. Therefore, their

proposed IDS responses are based on the assessed user risks which discount suspicious activities with a low risk that leads to an increase of resources availability. The major drawback in their designed IDS is that it is not robust enough to detect large scale (distributed) attacks since each IDS works independently. Takahashi et al. leveraged ontology and risk assessment approaches and introduced an ontological IDS on cloud computing which works as entity-based and it is equipped with a scoring system for vulnerabilities and weaknesses [29]. The proposed ontology recognizes three major factors: data-asset decoupling, composition of multiple resources and external resource usage which can be used as a set of common cyber-security terms and concepts in cloud computing.

A virtualization-based NIDPS for cloud computing environment was proposed by Jin et al. which used network data flow monitoring and real time file integrity [30]. Their proposed NIDPS had no control over the host which increased the vulnerability for insider attacks. As cost was always a major concern in developing CIDPS, Masud et al. formulated both of the malicious code detection and botnet traffic detection problems to introduce a new classification ensemble/integrated with machine learning technique which was a low-cost, scalable stream classification framework with high accuracy and low runtime overhead, but still suffers from high processing time in classification [31]. In a research by Dastjerdi et al., it was proposed to apply mobile agents in IDPS to provide flexible, scalable, and a cost effective system for the cloud environment [14]. However, they believed that this approach does not support enough robustness because of inefficient knowledge sharing between the mobile agents.

Table 1. Proposed CIDPSs for cloud computing classified according to our taxonomy

Ref.	Year	Detection technique	Technology layout	Detection time	Response type	Audit source	Management structure	Data diffusion	Prevention capability
[20]	2009	Hybrid - signature & anomaly	N/A	Real time	Active	Host & Network	Collaborative	Distributed	No
[14]	2010	N/A	Wireless; mobile agents	Real time	N/A	Network	Collaborative	Distributed	Yes
[29]	2010	Anomaly	N/A	Real time	Active	Network	Collaborative	Distributed	Yes
[25]	2010	Anomaly	N/A	Real time	Active	N/A	N/A	Distributed	No
[18]	2011	Hybrid - signature & anomaly	N/A	Real time	Active	Network	Individual	Distributed	Yes
[19]	2011	Anomaly	N/A	Real time	Active	Network	N/A	Distributed	No
[28]	2011	Anomaly	N/A	Real time	Active	Host & Network	Individual	Distributed	No
[22]	2011	Anomaly	N/A	Real time	Active	Host	Individual	Distributed	No
[30]	2011	Anomaly	N/A	Real time	Active	Network	Collaborative	Distributed	Yes
[23]	2012	Hybrid - signature & anomaly	Wireless; mobile agents	Real time	Active	Host & Network	Collaborative	Distributed	No

N/A = Not Applicable

Besides the available research on CIDPS, Zargar et al. presented a Distributed, Collaborative and Data-driven IDPS which works on three logical layers of network, host and global, in addition to platform and application levels. It maximizes the security and detection accuracy, since it monitors all operational changes and traffic movements which traverse through each layer. Their model provides *trust management* component among collaborative cloud providers to harmonize their respective IDPSs to ensure total synergized detection and protection [32].

Table 1 shows the most recent reviewed papers applicable to CIDPS, which are classified in terms of our proposed taxonomy. Their employed features are very similar to each other. The most important different features are *prevention capability, detection technique* and *system structure.*

5 CIDPS Architecture

The taxonomy presented in this research includes the advanced components for detection and prevention as shown as dashed boxes in Fig. 1. These advanced components use artificial intelligence techniques such as data mining, machine learning and fuzzy logic to detect intrusions and feed their results into the autonomic solution mode components comprising of self-healing, self-protecting, self-configuring, self-learning and self-optimizing in real-time without human intervention as defined in autonomic computing principals. The CIDPS proposed architecture of the system is illustrated and presented as a workflow scenario to show how it works in 10 steps as numbered in brackets in Fig. 2:

1. *Inputs from Autonomic Cloud Computing Environment Components:*
   Network, host, global, platform and applications are the autonomic cloud computing environment components. These components interactions generate and prepare the input sensor signals from the cloud environment. These signals, together with the latest CIDPS challenges and enterprise CIDPS policies and their updates, drive through the CIDPS Trust Management system to be analyzed.
2. *Latest CIDPS Challenges & Enterprise CIDPS Policies:*
   The CIDPS Enterprise Policies and Latest Challenges to cloud computing and their respective updates come into the CIDPS Trust Management system to complement the input sensor signals of autonomic cloud computing environment components as mentioned in Step 1. An incident entering the system is checked to determine if it is an intrusion or not. If it is an intrusion, then Intrusion Detection Engine (IDE) takes full responsibility to analyze and recognize the type of attack.
3. *Inference Engine (IE):*
   IE is the logical and main part of IDE. IE works based on the latest artificial intelligence techniques, fronted and equipped with a knowledge repository.
4. *Knowledge Repository:*
   This CIPDS architecture's Knowledge Base (KB) Repository (KBR) includes intrusion signatures, anomaly behavior patterns and policies. Given an intrusion/attack incident, KBR would be internally analyzed and updated, if necessary, automatically of a newly discovered intrusion incident by applying the set of AI techniques in each and every iteration of its execution cycle.

5. *Artificial Intelligence (AI) techniques:*
   Various AI techniques have been suggested in this architecture. Machine learning methods, data mining techniques, artificial neural network and fuzzy logic are the main AI techniques, which are proposed in this research. Artificial Neural Network (ANN) is used as a feature extraction selector and classifier of machine learning for IDE. The result of signal classification for intrusion detection would then be passed to the alarm management component.
6. *Alarm Management:*
   Alarm Management decides if the alarm trigger should be activated or not depending on a set of pre- and post-alarm criteria. If it is to be activated then fuzzy logic from Step 5 is employed to cluster the incident according to its severity and raise an alarm. The inference engine for IDS and alarm management components would access the knowledge repository via the AI techniques to retrieve the necessary event information for proper activation by targets further down the chain of components.
7. *Risk Assessment:*
   Risk Assessment prioritizes the intrusions according to the vulnerability of the victim. There are two possible cases in this state. The first case is if the incident is a severe intrusion, and the second possible case is related to the intrusions which are detected before any data loss or damage happens. It truly provides the impetus for the system to self-heal itself against any attacks, as well as, at the same time, caters for protection and prevention capabilities further down the chain in an autonomic mode of operation.
8. *Self-Healing Self-Protecting/Self-Preventing:*
   If the first case in Step 7 happens, it means that some parts have been already attacked or even infected. In this case, we may have some penetration tracks in our CIPDS Trust Management system and the cloud computing environment which would activate the self-healing component into action to ensure that the system protects itself. The second case refers to the detected and blocked intrusion before any data-loss happens. For this case, the system automatically enters the self-protecting state. In both cases, self-protecting/self-preventing state is triggered directly after risk assessment and self-healing is performed to protect the system by either using any one of the three automatic methods of self-configuring, self-learning and self-optimizing, or a combination of them.
9. *Self-Configuring, Self-Learning, Self-Optimizing:*
   These methods are triggered to protect the system by updating the CIDPS as a whole. Their actions are defined by the Inference Engine component in Trust Management. They would send signals to activate actuators to execute the *prevention* in the autonomic cloud computing environment components.
10. *Trust Management Actuators:*
    Trust Management Actuators carry and execute the defined actions of self-configuring, self-learning, and self-optimizing components. For instance, they reconfigure the victim's application settings; optimize the network traffic and policies; and even learn to respond with the correct behavior to the intrusions in the system. All of the decisions taken by the self-protecting and self-healing

components would go to the knowledge repository to be used for subsequent detections. This iterated workflow helps the whole trust management enterprise system to learn how to respond from previous incidents and experiences in their own environment.

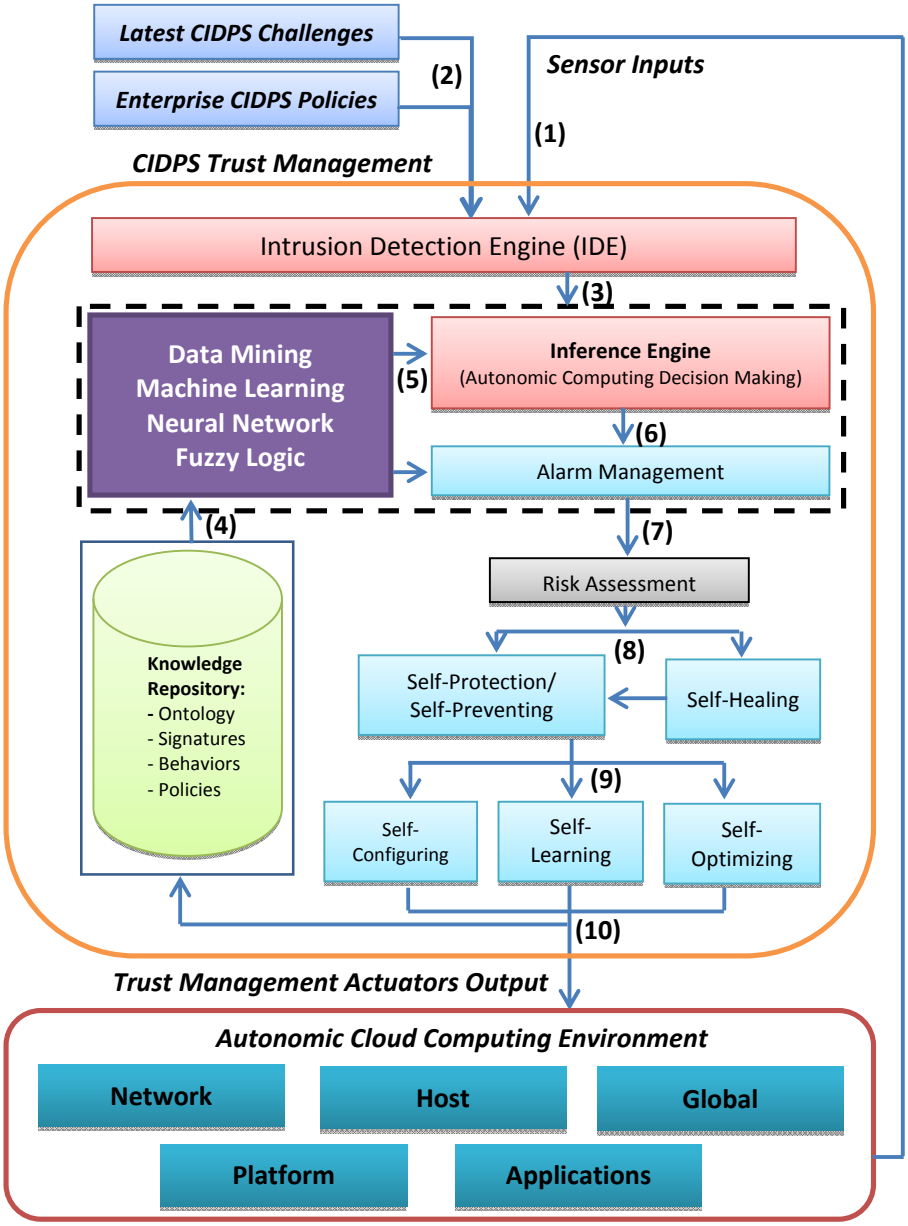

Fig. 2. CIDPS architecture with advanced components

6 Conclusion

This paper presented a comprehensive taxonomy and state of the art of intrusion detection and prevention systems to draw researchers' attention for possible solutions to intrusion detection and prevention in cloud computing. Among the reviewed papers, individual IDPS on each node increased the reliability of the system, but it exchanged higher traffic data over the network to synchronize the inter-operative nodes in the cloud environment, thus increasing processing time. Besides the structure of IDPS, detection technique was the other major factor that researchers paid a serious attention in their research. Anomaly and hybrid were the most common techniques discussed. Signature based system was faster because it only recognized the limited number of intrusions while anomaly learnt the traffic and actions to identify the safe activities and potential intrusions. The models which employed both types known as hybrid had the best accuracy and performance among the other individual methods. Monitoring dynamic virtual machines, scalability, minimizing human intervention and cost were the most important challenges to overcome by using advanced techniques and concepts of autonomic computing, ontology and risk assessment and analysis. There are still many issues unanswered which open research questions and doors for more investigation. Currently the proposed CIDPS Architecture with advanced techniques within the framework of autonomic computing principals is our primary research and development focus for cloud computing environments. We hope to implement, test and validate various intrusion detection algorithms and measure the effectiveness of the CIPDS architecture.

Acknowledgement. The authors thank the Ministry of Higher Education, Malaysia for supporting this research work through the Exploratory Research Grant Scheme (ERGS) number ERGS/1/2011/STG/UKM/01/16 and the Long Term Fundamental Research Grant Scheme (LRGS) number LRGS/TD/2011/UKM/ICT/02/01 projects.

References

[1] Cloud-Security-Alliance, Top Threats to Cloud Computing V1.0 (2010), https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf

[2] Galante, J., Kharif, O., Alpeyev, P.: Sony Network Breach Shows Amazon Cloud's Appeal for Hackers (2011), http://www.bloomberg.com/news/2011-05-15/sony-attack-shows-amazon-s-cloud-service-lures-hackers-at-pennies-an-hour.html

[3] Shabtai, A., Fledel, Y., Kanonov, U., Elovici, Y., Dolev, S., Glezer, C.: Google android: A comprehensive security assessment. IEEE Security & Privacy 8, 35–44 (2010)

[4] Bringas, P.G., Penya, Y.K.: Next-Generation Misuse and Anomaly Prevention System. In: Filipe, J., Cordeiro, J. (eds.) ICEIS 2008. LNBIP, vol. 19, pp. 117–129. Springer, Heidelberg (2009)

[5] Elshoush, H.T., Osman, I.M.: Alert correlation in collaborative intelligent intrusion detection systems—A survey. Applied Soft Computing 11, 4349–4365 (2011)
[6] Klüft, S.: Alarm management for intrusion detection systems - Prioritizing and presenting alarms from intrusion detection systems. MSc Thesis, University of Gothenburg (2012), http://hdl.handle.net/2077/28856
[7] Lippmann, R., Webster, S., Stetson, D.: The Effect of Identifying Vulnerabilities and Patching Software on the Utility of Network Intrusion Detection. In: Wespi, A., Vigna, G., Deri, L. (eds.) RAID 2002. LNCS, vol. 2516, pp. 307–326. Springer, Heidelberg (2002)
[8] Maggi, F., Matteucci, M., Zanero, S.: Reducing false positives in anomaly detectors through fuzzy alert aggregation. Information Fusion 10, 300–311 (2009)
[9] Leitner, M., Leitner, P., Zach, M., Collins, S., Fahy, C.: Fault management based on peer-to-peer paradigms; a case study report from the celtic project madeira. In: 10th IFIP/IEEE International Symposium on Integrated Network Management, pp. 697–700 (2007)
[10] Zhou, C.V., Leckie, C., Karunasekera, S.: A survey of coordinated attacks and collaborative intrusion detection. Computers & Amp; Security 29, 124–140 (2010)
[11] Arshad, J., Townend, P., Xu, J.: A novel intrusion severity analysis approach for Clouds. Future Generation Computer Systems (2011), http://dx.doi.org/10.1016/j.future.2011.08.009
[12] Grobauer, B., Walloschek, T., Stocker, E.: Understanding cloud computing vulnerabilities. IEEE Security & Privacy 9, 50–57 (2011)
[13] Viega, J.: Cloud computing and the common man. Computer 42, 106–108 (2009)
[14] Dastjerdi, A.V., Bakar, K.A., Tabatabaei, S.G.H.: Distributed intrusion detection in clouds using mobile agents. In: Third International Conference on Advanced Engineering Computing and Applications in Sciences, Sliema, pp. 175–180 (2009)
[15] Wang, C., Wang, Q., Ren, K., Lou, W.: Ensuring data storage security in cloud computing. In: 17th International Workshop on Quality of Service (IWQoS 2009), Charleston, SC, pp. 1–9 (2009)
[16] Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud computing and grid computing 360-degree compared. In: Grid Computing Environments Workshop, GCE 2008, Austin, TX, pp. 1–10 (2008)
[17] Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications 34, 1–11 (2011)
[18] Tupakula, U., Varadharajan, V., Akku, N.: Intrusion Detection Techniques for Infrastructure as a Service Cloud. In: IEEE International Conference on Dependable, Autonomic and Secure Computing, pp. 744–751 (2011)
[19] Gustavo, N., Miguel, C.: Anomaly-based intrusion detection in software as a service. In: Dependable Systems and Networks Workshops, pp. 19–24 (2011)
[20] Vieira, K., Schulter, A., Westphall, C.: Intrusion Detection for Grid and Cloud Computing. IT Professional 12, 38–43 (2010)

[21] Xin, W., Ting-lei, H., Xiao-yu, L.: Research on the Intrusion detection mechanism based on cloud computing. In: 2010 International Conference on Intelligent Computing and Integrated Systems (ICISS), Guilin, pp. 125–128 (2010)
[22] Dhage, S., Meshram, B., Rawat, R., Padawe, S., Paingaokar, M., Misra, A.: Intrusion detection system in cloud computing environment. In: International Conference & Workshop on Emerging Trends in Technology, New York, NY, USA, pp. 235–239 (2011)
[23] Kholidy, H.A., Baiardi, F.: CIDS: A Framework for Intrusion Detection in Cloud Systems. In: Ninth International Conference on Information Technology: New Generations (ITNG), Las Vegas, NV, pp. 379–385 (2012)
[24] Patel, A., Qassim, Q., Shukor, Z., Nogueira, J., Júnior, J., Wills, C.: Autonomic Agent-Based Self-Managed Intrusion Detection and Prevention System. In: South African Information Security Multi-Conference (SAISMC 2010), Port Elizabeth, South Africa, pp. 223–224 (2009)
[25] Smith, D., Guan, Q., Fu, S.: An Anomaly Detection Framework for Autonomic Management of Compute Cloud Systems. In: 34th Annual Computer Software and Applications Conference Workshops (COMPSACW), Seoul, pp. 376–381 (2010)
[26] Martínez, C.A., Echeverri, G.I., Sanz, A.G.C.: Malware detection based on cloud computing integrating intrusion ontology representation. In: IEEE Latin-American Conference on Communications (LATINCOM), Bogota, pp. 1–6 (2010)
[27] Azmandian, F., Moffie, M., Alshawabkeh, M., Dy, J., Aslam, J., Kaeli, D.: Virtual machine monitor-based lightweight intrusion detection. SIGOPS Oper. Syst. Rev. 45, 38–53 (2011)
[28] Lee, J.H., Park, M.W., Eom, J.H., Chung, T.M.: Multi-level Intrusion Detection System and log management in Cloud Computing. In: 13th International Conference on Advanced Communication Technology (ICACT), Seoul, pp. 552–555 (2011)
[29] Takahashi, T., Kadobayashi, Y., Fujiwara, H.: Ontological approach toward cybersecurity in cloud computing. In: 3rd International Conference on Security of Information and Networks, Taganrog, Rostov-on-Don, Russian Federation (2010)
[30] Jin, H., Xiang, G., Zou, D., Wu, S., Zhao, F., Li, M., Zheng, W.: A VMM-based intrusion prevention system in cloud computing environment. The Journal of Supercomputing, 1–19 (2011)
[31] Masud, M.M., Al-Khateeb, T.M., Hamlen, K.W., Gao, J., Khan, L., Han, J., Thuraisingham, B.: Cloud-based malware detection for evolving data streams. ACM Trans. Manage. Inf. Syst. 2, 1–27 (2008)
[32] Zargar, S.T., Takabi, H., Joshi, J.B.D.: Dcdidp: A Distributed, Collaborative, and Data-Driven Intrusion Detection and Prevention Framework for Cloud Computing Environments. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing CollaborateCom, Orlando, Florida, USA (2011)