Journal of Network and Computer Applications 36 (2013) 25–41

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

An intrusion detection and prevention system in cloud computing: A systematic review

Ahmed Patel [a,b], Mona Taghavi [a,*], Kaveh Bakhtiyari [a], Joaquim Celestino Júnior [c]

[a] School of Computer Science Centre of Software Technology and Management (SOFTAM), Faculty of Information Science and Technology (UKM), Universiti Kebangsaan Malaysia (UKM), 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia

[b] Visiting Professor School of Computing and Information Systems, Faculty of Science, Engineering and Computing, Kingston University, Kingston upon Thames, KT1 2EE, United Kingdom

[c] Vieira Computer Networks and Security Laboratory (LARCES), State University of Ceará (UECE), Fortaleza, Ceará, Brazil

ARTICLE INFO

Article history:
Received 24 April 2012
Received in revised form
27 July 2012
Accepted 23 August 2012
Available online 1 September 2012

Keywords:
Intrusion detection and prevention
Cloud computing
Taxonomy
Alarm correlation
System requirements

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 largely inefficient to be deployed in cloud computing environments due to their openness and specific essence. This paper surveys, explores and informs researchers about the latest developed IDPSs and alarm management techniques by providing a comprehensive taxonomy and investigating possible solutions to detect and prevent intrusions in cloud computing systems. Considering the desired characteristics of IDPS and cloud computing systems, a list of germane requirements is identified and four concepts of autonomic computing self-management, ontology, risk management, and fuzzy theory are leveraged to satisfy these requirements.

Contents

1. Introduction

Over the last decade, our society has become technology dependent. People rely on computer networks to receive news, stock prices, email and online shopping. The integrity and availability of

* Corresponding author.
E-mail addresses: whinchat2010@gmail.com (A. Patel), mona.taghavi@gmail.com (M. Taghavi), academic@bakhtiyari.com (K. Bakhtiyari), jcelestinojr@gmail.com (J. Celestino Júnior).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.08.007

all these systems need to be defended against a number of threats. Amateur hackers, rival corporations, terrorists and even foreign governments have the motive and capability to carry out sophisticated attacks against computer systems (Choo, 2011). Therefore, the field of information security has become vitally important to the safety and economic well being of society as a whole. The rapid growth and widespread use of electronic data processing and electronic business conducted through the massive use of the wired and wireless communication networks, Internet, Web application, cloud computing along with numerous occurrences of international terrorism, raises the need for providing secure and safe information security systems through the use of firewalls, intrusion detection and prevention systems, encryption, authentication and other hardware and software solutions.

In this struggle to secure our stored data and the systems, 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 (Shabtai et al., 2010). By using IDPS, one can potentially identify an attack and notify appropriate personnel immediately or prevent it from succeeding, so that the threat can be contained. IDPS can also be a very useful tool for recording forensic evidence that may be used in legal proceedings if the perpetrator of a criminal breach is prosecuted (Sy, 2009).

However, IDPS performance is hindered by the high false alarm rate it produces (Wu and Banzhaf, 2010). This is a serious concern in information security because any false alarms will onset a severe impact to the system such as the disruption of information availability because of IDPS blockage in suspecting the information to be an attack attempt.

1.1. Research motivation

The fully distributed and open structure of cloud computing and services becomes an even more attractive target for potential intruders. It involves multi-mesh distributed and service oriented paradigms, multi-tenancies, multi-domains, and multi-user autonomous administrative infrastructures which are more vulnerable and prone to security risks. Cloud computing service architecture combines three layers of inter-dependent infrastructure, platform and application; each layer may suffer from certain vulnerabilities which are introduced by different programming or configuration errors of the user or the service provider. A cloud computing system can be exposed to several 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 (Cloud-Security-Alliance, 2010). The problem becomes even more critical when a cloud with massive computing power and storage capacity is abused by an insider intruder as an ill-intention party which makes cloud computing a threat against itself.

Around last year (2011), a hacker used Amazon's Elastic Computer Cloud service to attack Sony's online entertainment systems by registering and opening an Amazon account and using it anonymously (Galante et al., 2011). Cloud services are as cheap and convenient for hackers as 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.

Lack of full control over the infrastructure is a major concern for the cloud services' consumers. It signifies the role of IDPS in protecting the users' information assets in cloud computing.

This research amalgamates the challenges and issues banning further development of advanced IDPSs in a cloud computing environment. It aims to attract well-respected researchers' attention to possible solutions of developing IDPSs by bringing the latest disparate research works together to shed light on securing the recent widespread cloud services and resources. Besides, it identifies desired IDPS requirements for cloud computing and suggests important implications for practice.

1.2. Research boundaries & limitations

Among the existing solutions for IDPS, all-purpose systems which consider more aspects of the solution and components of the system are studied in this paper. Some of the researchers focused on a particular IDPS component or a specific type of attack or intrusion, targeted to decrease false-positive rates. For example, Carl et al. investigated different denial-of-service (DoS) detection techniques (Carl et al., 2006), and Sixsmith & Johnson tried to improve the quality of sensors to detect more intrusions (Smith and Johnson, 2004).

Given that cloud computing is the target environment; this research focuses more attention to recent research works which were published in the last few years to take advantage of the advanced and up-to-date systems. Prevention is a newly acquired feature for intrusion detection systems, thus, there are only a few published research papers including this feature. Nevertheless, this study considers all recent works on intrusion prevention systems (IPS) or intrusion detection systems (IDS) since they are expandable by adding a prevention module. These restrictions limit this research with the current state of the art.

The analysis of these research works is qualified based on the provided taxonomy which is explained in the following section. Cloud computing is only recently adopted worldwide; therefore, there are very few practical and experimental intrusion detection systems developed in the real world. Regardless of all the boundaries and limitations of the current state of the art, this research is based on the following two questions:

1. What criteria and requirements should an IDPS meet to be deployed on cloud computing environments?
2. Which methods or techniques can satisfy these requirements?

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 (Whitman and Mattord, 2011). 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.

An IDS is a software that automates the intrusion detection process and detects possible intrusions. An IDPS is a software or hardware device that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents. IPSs are differentiated from IDSs by one characteristic; IPS can respond to a detected threat by attempting to prevent it from succeeding (Scarfone and Mell, 2007). The IPS changes the attack's content and/or changes the security environment. It could change the configuration of other security controls to disrupt an attack, such as reconfiguring a network device to block access from the attacker or to the victim, or altering a host-based firewall on a target to block incoming attacks. Some IPSs can remove or replace malicious portions of an attack to make it benign. Because of the high false alarm rates of the anomaly detection (Patel et al., 2010), IPS incorrectly identifies a legitimate non-intrusive normal activity as malicious and responds to that detected activity inaccurately.

Fig. 1 shows a traditional IDPS in the blue box which is working within a cloud by collecting the audit data, analyzing the data and detecting the intrusion, generating an alarm and proceeding with the proper response. This process is shown only for one of the entities while it needs to be performed consistently for all of the entities to protect the cloud resources from malicious activities. The advanced components which are shown in the red box will be explained in Section 5.

Fig. 2 provides high level taxonomy of IDPSs. The explanations of the taxonomy elements are discussed as below.

2.1. Functional layer

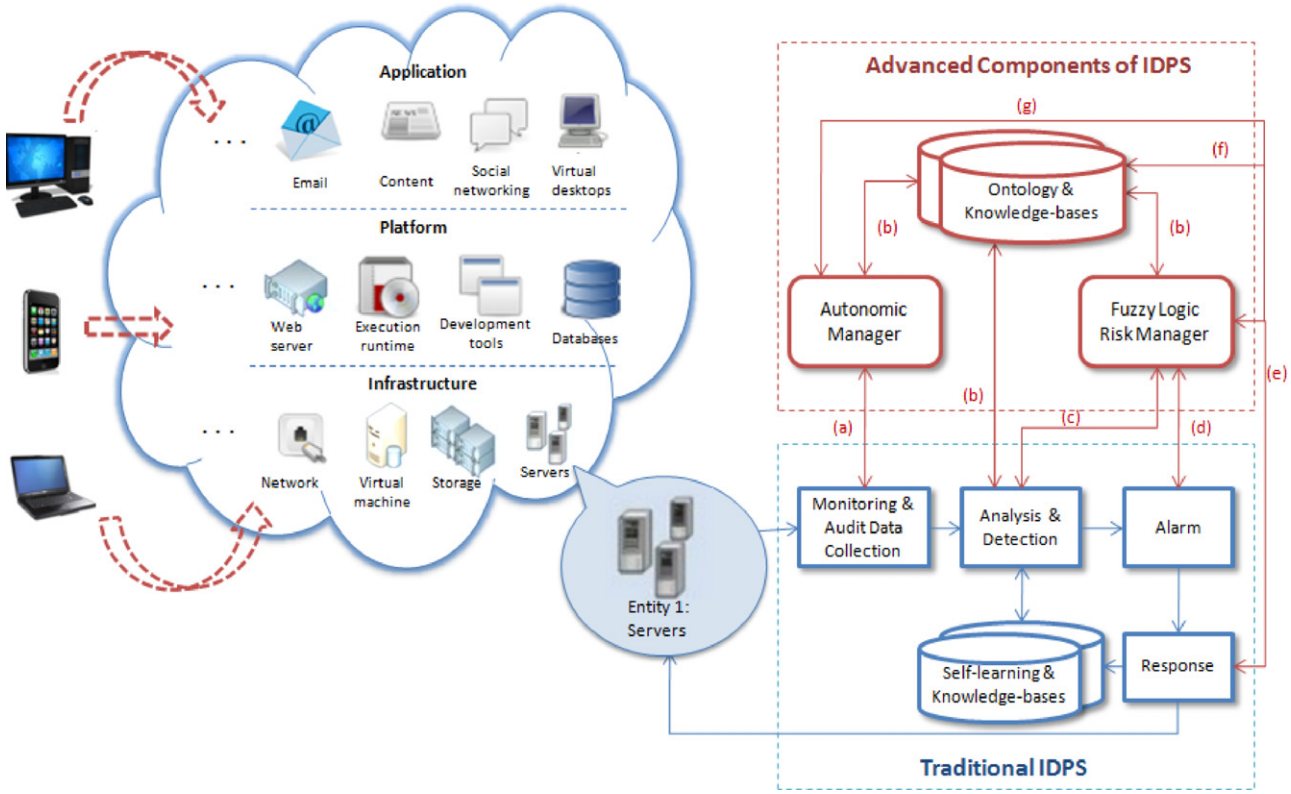
As Fig. 2 shows, IDPSs serve four essential security functions: they monitor, detect, analyze, and respond to unauthorized activities as depicted in the functional layer.

An IDPS detects intrusion by analyzing the collected data. The monitored environment can be network-based, host-based or application-based:

1. Network-based (NIDPS): monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity.
2. Host-based (HIDPS): monitors all or parts of 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 both network-based and host-based components which provides greater flexibility in deployment.
3. Application-based (AIDPS): concentrates on events which occur in some specific application through analyzing the application log files or measuring their performance. Its input is data sources of running applications.

In real-time detection, attacks are identified while the system or network is being monitored for intrusions and can immediately flag any deviations and provide proper prevention. The real-time IDPS can also be run for off-line analysis through historical data to identify past intrusions. By contrast, a non-real-time IDPS processes audit data with delay. Audit data can be collected in a distributed fashion from several different locations or sources, or they can be collected in a centralized approach from one single source.

The identified methods of detection are classified in three classes of misuse, anomaly and hybrid model combining the first two classes:



Legend:
a) Autonomic manager monitor and to configure the IDPS according to hardware and software changes.
b) Ontology unifies knowledge-base to facilitate information sharing.
c) Fuzzy logic risk manager allows system to calculate and analyze vulnerabilities and risks.
d) Fuzzy logic risk manager controls triggering false positive alarms.
e) System response based on risk severity and impact calculated by fuzzy logic risk manager
f) Response will be recorded in knowledge base for future attacks. Ontologies allow synchronizing autonomous agents.
g) Autonomic manager helps system to optimize its use of resources and provide real-time response without human intervention.

Fig. 1. A typical IDPS for one of the entities within cloud computing.

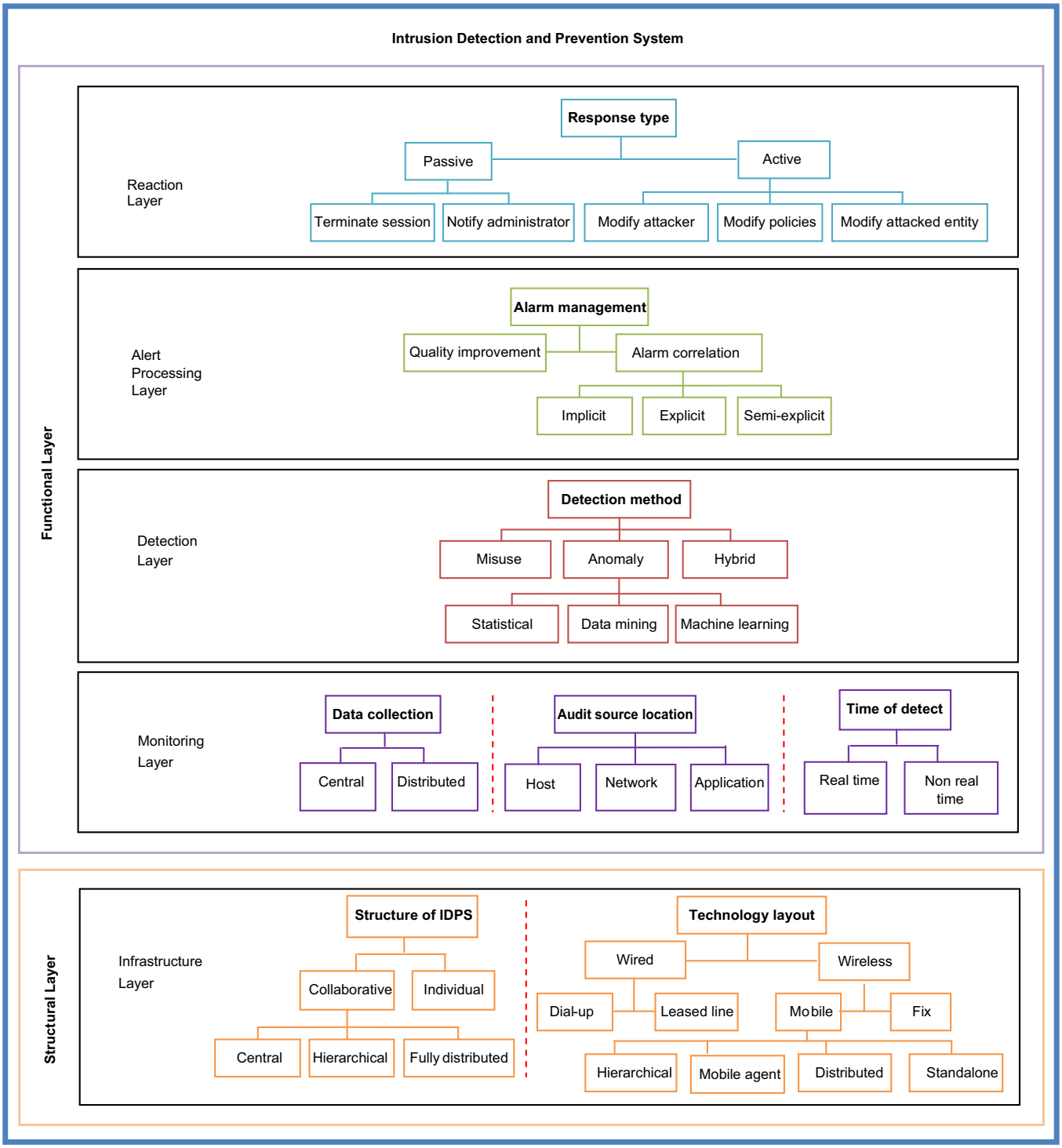


Fig. 2. A layered-taxonomy of IDPS.

1. Misuse detection: this method uses specifically known patterns of unauthorized behavior, called signatures, to predict and detect subsequent similar attempts.
2. Anomaly detection: designed to uncover abnormal patterns of behavior. IDPS establishes a baseline of normal usage patterns, and whatever deviates from this get flagged as possible intrusions (Thatte et al., 2011). What is considered to be an anomaly can vary, but normally, any incident that occurs on frequency greater than or less than two standard deviations from the statistical norm raises an eyebrow (Bringas and Penya, 2009). There are various categories of anomaly detection proposed, but the three most used ones are as follows (Hoang et al., 2009; Elshoush and Osman, 2011):

- Statistical: in this approach the system observes the activity of subjects (such as CPU usage or number of TCP connections) in terms of statistical distribution and creates profiles which represent their behaviors. Therefore, 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 based: this technique has the ability of learning and improving its performance over time. It tends

to focus on constructing a system which can optimize its performance in a loop cycle and can change its execution strategy according to feedback information. System call-based sequence analysis, Bayesian network and Markov model are the most frequently used techniques.
- Data mining based: Data mining techniques can help to improve the process of intrusion detection by unfolding patterns, associations, anomalies, changes, and important events and structures in data. Classification, clustering and outlier detection, and association rule discovery are data mining techniques used in IDPS.

3. Hybrid: this approach has been proposed to enhance the capabilities of a current 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 categorized into two methods (Pietraszek and Tanner, 2005; Klüft, 2012):

1. Alert (alarm) quality improvement: this approach tries to improve the alert quality by using additional information, such as vulnerability reports or alert context. There is also another approach which matches vulnerability reports with correlated alerts. Lippmann et al. (2002) prioritized alerts according to the vulnerabilities of the victim in a way that correctly identified intrusions were given lower prioritization or discarded if that specific victim is not vulnerable to that attack .
2. Alarm correlation: this approach follows a more ambitious goal. It tries to reconstruct 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. Alarm correlation can be performed in three ways:
- Implicit: it uses data-mining techniques in order to analyze, aggregate and group large alert datasets. However, this method fails to enhance the semantics of the alerts, but is suitable for analysis of huge number 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 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 post-conditions of some prior ones.

When an IDPS responds actively to an intrusion, it may further modify the attacked system state or, in rare cases, modify the attacker state by removing his platform. In some cases, they can instruct network security devices to reconfigure themselves to block certain types of activity or route it elsewhere. They might reconfigure network firewalls by changing user access control policy temporarily as an attack occurs. Passive systems can attempt to terminate the connection before an attack can succeed, for example, by ending an existing TCP session.

2.2. Structural layer

Referring to Fig. 2, the technology of an IDPS is located in the infrastructure layer. 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 line or leased which is analog compatible point to point connection. In wired networks, the features like traffic behavior and network topology can be employed in detecting of intrusions (Estevez-Tapiador et al., 2004). A mobile ad-hoc network is a collection of mobile nodes that automatically self-configure without assistance of a central management of infrastructure. The wireless network IDPS 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 IDPS is achieved by physically integrating it within a firewall. 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: detection element 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. As Fig. 3 shows, collaborative IDPS can be divided in three categories as follows Elshoush and Osman (2011):

1. *Central*: each IDPS acts as a detection element where it produces alerts locally. The generated alerts will be 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 an 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.

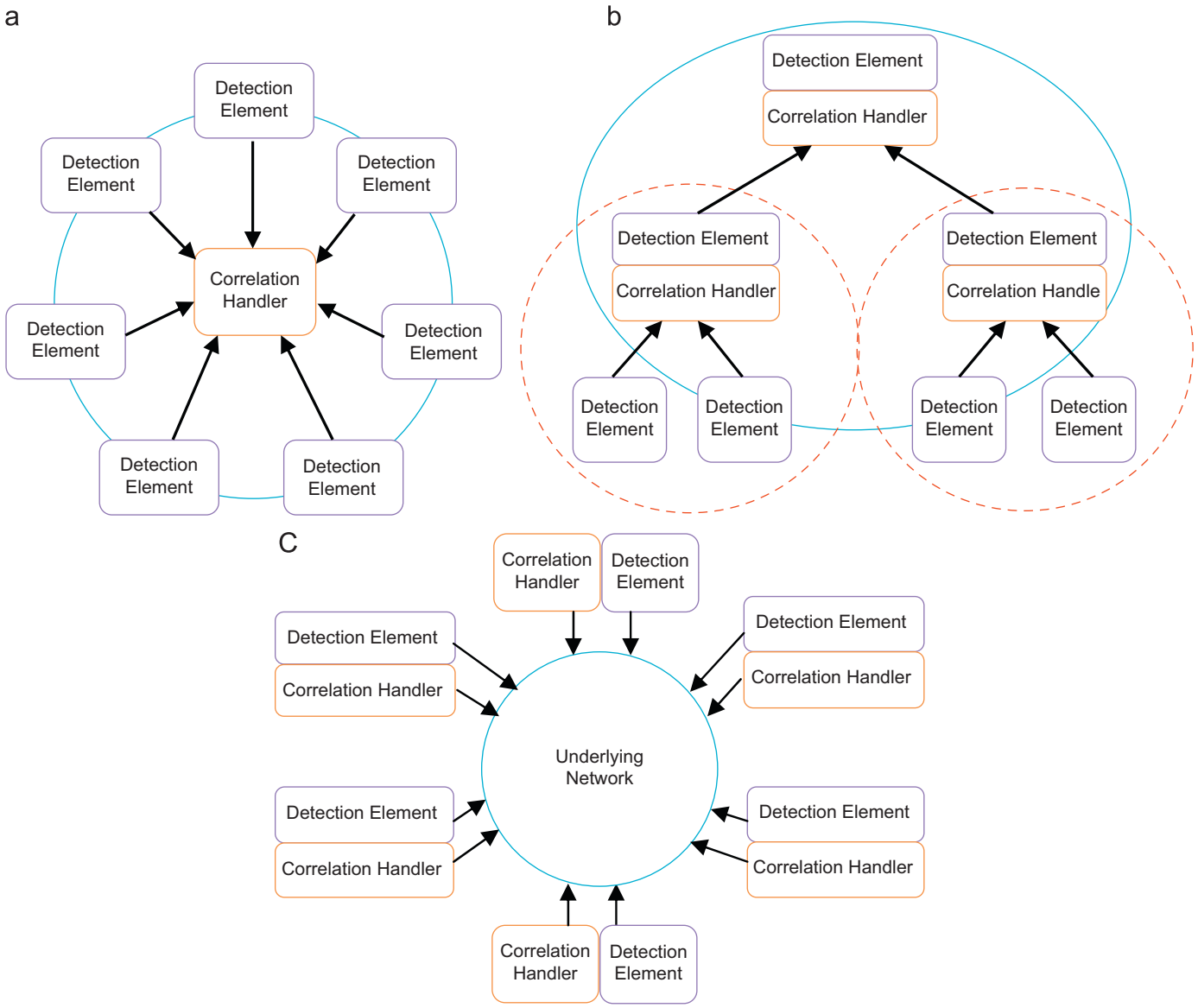


Fig. 3. Different management structures of collaborative IDPSs. (a) Central, (b) Hierarchical and C) Fully Distributed.

3. *Fully distributed*: there is no centralized coordinator to process the information, it compromises fully autonomous systems with distributed management control. All participant IDPSs have their own two main function components communicating with each other. The advantages of the fully distributed IDPS include (Leitner et al., 2007): the network entities has not to have complete information of the network topology, it is possible to have a more scalable design since there is no central entity responsible for doing all the correlation works, and local alarm correlation is simpler in this structure. Meanwhile, fully distributed approach has its own drawback issues (Zhou et al., 2010): (a) the information of all alerts is not available during the detection decision making, so the accuracy might reduce; (b) the alert information usually has a single feature (like an IP address) which is too narrow for detecting large scale attacks.

3. Current state of the art of IDPS

Latest research findings on IDPSs are organized in terms of layered taxonomy and discussed in two parts. The first part provides an insight view on recent proposed systems in terms of their structure, technology, processes of audit data collection and analysis, detection techniques, and types of responses. The second part only targets the research studies on alarm management which have focused on false positive alarm reduction by applying various methods for different IDPS detection techniques.

3.1. Intrusion detection and prevention systems

Table 1 provides a comprehensive list of the most recent proposed IDPS based on the defined layered taxonomy in the previous section. What is obvious from Table 1 is that recent research has concentrated more on collaborative systems to provide solutions for distributed real-time environment using hybrid detection techniques and wireless technologies. However, many of the researchers have provided a solution to a specific problem and did not try to optimize the whole system in terms of the proposed taxonomy components. For example, the detection technique with high accuracy is one of the most favorable research areas regardless of other consequent challenges such as false alarm rate and response time or type.

Table 1
Classification of existing IDPSs based on a layered-taxonomy.

Reference	Year	Detection technique	Technology layout	Time of detect	Response type	Audit source location	Management structure	Data diffusion	Remarks: prominent advantage or disadvantage
Khanum et al. (2012)	2012	Signature based	Wireless (mobile)	Real time	Passive	Network	Collaborative	Distributed	Utilize minimum possible network resources.
Chung-Ming (2012)	2012	Artificial immune system	Wireless (mobile-agent)	Real time	Passive	Host	Collaborative	Distributed	Capable of fast detection but weak in intrusion severity, certainty.
Jaiswal and Jain (2010)	2010	Genetic Algorithm	Wired	Real time	Active	Host	Individual	Centralized	The different types of attacks for database are considered.
Vieira et al. (2010)	2010	Hybrid (signature and anomaly)	Wireless	Real time	Passive	Application	Collaborative	Distributed	Low computational cost.
(Li et al. (2010)	2010	Immune system (Dynamic clonal selection algorithm)	Wireless (mobile-agent)	Real time	Active	Network	Collaborative (fully distributed)	Distributed	Ability to deal with a high-volume network traffic data stream.
(Awodele et al. (2009)	2009	Hybrid (signature and anomaly)	Wired	Real time	Active	Host	Individual	Centralized	Covers only a single host.
(Shyu and Sainani (2009)	2009	Data mining (supervised classification)	Wireless (mobile-agent)	Real time	Active	Network	Collaborative	Distributed	Linear scalability and low response time.
(Rasoulifard et al. (2009)	2009	Hybrid (incremental misuse detection and incremental anomaly detection)	unspecified	Real time	Passive	Network	Individual	Distributed	Low computational complexity
(Su et al. (2009)	2009	Fuzzy association rules	Wireless	Real time	Passive	Network	Individual	Centralized	Fewer false alarms.
(Herrero and Corchado (2009)	2009	Case-based reasoning and an unsupervised neural projection model	Wireless (mobile-agent)	Real time	Passive	Network	Collaborative	Distributed	Ability to deal with a high-volume network traffic data stream.
(Byrski and Carvalho (2008)	2008	Immune-inspired and agent based	Wireless	Real time	Passive	Network	Collaborative	Distributed	Independent of specific routing protocols and services.
(Sproull and Lockwood (2007)	2007	Hybrid (signature and anomaly)	Wired	Real time	Active	Network	Collaborative	Distributed	Has minimal impact on overall network performance.
(Liang et al. (2006)	2006	Hybrid (immune system)	Unspecified	Real time	Active	Network	Unspecified	Unspecified	Easy to adapt into a dynamic changing network environment.

In addition to neglecting the whole system requirements, the proposed IDPSs suffer from several issues. The evolving challenges which limit IDPS development (in particular for anomaly based systems) are as follows:

1. Traditional IDPSs have not applied adequately to new networking paradigms like mobile and wireless networks. They also failed to be scaled to satisfy the high-speed networks requirements (Patcha and Park, 2007).
2. The traffic profiles regularly change due to some negative factors (like noise) in the audit data that make it difficult to build a normal traffic profile in a large amount of network traffic.
3. One of the most serious constraints factors blocking the widespread usage of IDPS is their grossly high produced false alarm rate Perdisci et al., 2006.
4. In spite of the numerous proposed techniques, models and implemented systems (even commercially), there is no uniform globally accepted standard or metric to evaluate an IDPS, although receiver operating characteristic (ROC) has been widely used for accuracy evaluation, but they are far from a desirable assessment tools due to their often incomplete and mislead evaluation result (Gaffney Ulvila, 2011).
5. It is very difficult to detect internal attacks, meanwhile the insiders' threats are increasing (Moore et al., 2008). Proper configuration of the system and provision of suitable policies and rule sets for internal intruders are very challenging tasks.

A comparative analysis of different taxonomy components are provided in Table 2. This Table lists the advantages and disadvantages of each feature of an IDPS. However, among the various solutions and techniques the most applied and significant ones are discussed.

Among all the features, detection techniques are in the center of attraction. Through the review of existing surveys (García-Teodoro et al., 2009; Nazer and Selvakumar, 2011; Xie et al., 2011), a list of criteria was collected to compare the detection techniques which are based on signature, anomaly or hybrid of these two methods. Table 3 provides this comparative review based on the collected criteria.

As shown in Table 3, hybrid methods inherit advantages of both anomaly and misuse detections and cover deficiency holes of each technique.

A number of desired characteristics are identified for an ideal IDPS to have optimized performance, maximum protection and minimum error (Sharma and Sinha, 2011; Patel et al., Wills):

IC1: Run continuously with minimum or without human supervision.
IC2: Be survivable and fault tolerant to be able to recover when system crashes.
IC3: Be simply tailored to a particular network.
IC4: Adapt to changes in user behavior and system over time.
IC5: Work in real-time.
IC6: Recognize all or most intrusions with minimum number of false-positive alarms.
IC7: Be self-monitored and self-protected in case it is modified by an attacker.
IC8: Be self-configurable according to the changing security policies of the system under supervision.
IC9: Operate with minimum overhead while system is running.

Table 2
Comparative analysis of ID/PS features.

Features	Advantages	Disadvantages
Technology layout		
• Wired • Wireless	• Wired networks are faster and low cost. • It offers wide coverage and unlimited access which implicate openness to attacks. • It is scalable and independent from infrastructure platform. • Mobile agents has less energy consumption Chen et al., 2006.	• They are heavily dependent on structure platform and not easy to deploy. • In addition to attacks that may be performed on a wired network, the wireless medium itself has to be protected.
Detection method		
Misuse	• Misuse detectors are reliable, efficient and generate a very low false alarm rate in detecting specified and well-known intrusions (Hoang et al., 2009).	• Misuse detection is showing its severe limitation in unknown attacks detection (called *zero-days*) as new attacks are constantly evolving. Their inability is not only limited to unknown attacks, they have difficulty for even intrusions which are already known as attacks but have unknown signatures. The probability of erroneously misclassification of normal events as attacks is high. Since each event should be compared against many signatures using computational resources, it leads to reducing detection rate and overall performance. False alarms may are derived from poorly constituted signatures. They might become a single point of failure (Shon et al., 2006). If intrusion systems become disabled for any reason, it gives time to an attacker to compromise the systems. Matching the signatures are well done for single connection attacks only, while most of the attacks involve multiple connections (Hwang et al., 2007).
Anomaly	• Anomaly based techniques use fewer rules to the signature based techniques, this increases detection rate and effectiveness. • It is able to detect most new attacks without a need to be updated because a new attack deviates from protocol specifications.	• Anomaly detectors have higher false positive alarms, because deviating from normal behavior does not always mean that an attack is occurring. • It is too difficult to discover the boundaries between abnormal and normal behavior. The deficiency of abnormal samples in the training phase challenge defining the normal behavior (Chandola et al., 2009). • Another difficulty exists in adapting to continuously changing normal behavior, particularly for dynamic anomaly. • Attacker can change the behavior patterns so that it will accept attack behavior as normal.
Time of detection		
• Real-time • Non real-time	• It excels the progress of attacks detection and prevention. • It can fill the network inherent security gaps associated with vulnerability to various types of attacks (especially DoS) that are not detectable by common approach of audit trail analysis (Kazienko and Dorosz, 2004). • It has high capabilities to provide evidence of data forensic. • It has less resource consumption	• Real-time detection cannot handle encrypted packets, so they are not able to provide essential information which is required for intrusion detection. • The performance of the real-time system is affected by a running agent through the system. • Source identification is achieved based on the network address from the packet (not for example using the network ID). Therefore, the source address may be spoofed and makes it hard to trace and responds attacks automatically (Axelsson, 2000). • It cannot provide real time response to prevent or mitigate damages.
Data		
• Distributed • Central	• The distributed data utilizes the traffic information from various sources in the form of data to investigate the security status of its residing network. • All of the monitoring, detection, and response activities are controlled directly by a central console.	• The data flow between host monitors and the director agent may generate significantly high network traffic overheads. • The information used by the system is mainly obtained from packets or from audit trails on a network. So, data have to traverse a longer path from their origin to the intrusion system, and in the process can potentially be destroyed or modified by an attacker which may result in misinterpretations or missed events (Kerschbaum et al., 2002). • An intruder can modify or disable the programs running on a system, making the IDPS useless or unreliable.
Audit source location		
• Network based • Host based	• NIDSs are able to detect attacks that host-based systems miss since they monitor network traffic at the transport layer. • Their strategic position allows for quick response. • They are able to verify success or failure of an attack quickly because they log events that have actually occurred continuously, they have information that is more accurate and less prone to false positives than HIDSs. • They are easier to deploy as it does not affect existing infrastructures. • Since they monitor a local host, they are able to see low-level local activities such as file accesses, changes to file permissions. • HIDSs can deal with encrypted and switched environments. • They are cost effective. • HIDSs do not require additional hardware.	• NIDSs are far from individual hosts, thus they are not aware of implementation of each host's protocol. • Since NIDSs do not have a full picture of the network topology between the NIDS and the hosts, the NIDS may not be able to determine a given packet received by the hosts (Kizza, 2009). • They have no capability to decrypt encrypted data. They only can scan unencrypted parts of the packet such as headers (Kizza, 2009). • Since NIDSs are on dedicated machines routinely protected, they have difficulty to remove evidence. • Their deployment at a host causes a very limited view of the network. • It can help to detect a Trojan horse or other attacks that involve software integrity breaches.

Table 2 (*continued*)

Features	Advantages	Disadvantages
Response type		
• Passive • Active	• It facilitates the flow of information by allowing alarm events to access the information assets • Active response blocks alarm events immediately to protect information assets.	• Passive response exposes the assets to attacks while the security administrator investigates the alarms. It may delay the benign traffic unnecessarily since alarm events are blocked. • The optimal configuration under active response is smaller. than an under passive response (Yue and Çakanyıldırım, 2010).
Alarm management		
• Quality improvement • Alarm correlation	• This approach is simple to implement and is adoptable to most of current alert correlation systems (Pietraszek and Tanner, 2005). • This approach is intuitive and effective in real environments	• Using this approach individually is inefficient and does not provide an optimized solution to minimize false positive alarms. • Most of the algorithms proposed in the current literature on correlation make use of the matching attack information provided by misuse detectors (Maggi et al., 2009)
Structure		
• Individual • Collaborative	• It is easy to deploy due to its independent structure. • Collaborative IDPSs are more efficient to detect and prevent intrusions over the Internet (Zhou et al., 2010). • They may reduce computational costs by sharing intrusion detection resources between networks (Zhou et al., 2010). • They provide comprehensive information about intrusion attempts for alarm correlation purposes.	• It produces more irrelevant and false alarms (Elshoush and Osman, 2011). • For a specific attack, only some of IDPSs might be able to detect it (Perdisci et al., 2006). • Can have different outputs from different IDPSs for an attack. • Due to their distributed architecture they have less scalability • Different detection techniques need different computation power and speed, so in an event may slow a IDPS generated alarm after the others.

Table 3
Comparison of detection methods based on collected criteria from existing surveys (García-Teodoro et al., 2009; Nazer and Selvakumar, 2011; Xie et al., 2011) and taxonomy.

Comparison c criteria Detection techniques	Robustness	Flexibility	Scalability	Resource consuming	Alarm rate	Reliability	Speed	Commercial tools
Signature	Low	Low	Low	Low	Low	High	High	Cisco NetRanger, Snort, Nessus
Anomaly	High	High	High	High	High	Midrate	Low	Mazu profiler, nPatrol, SPADE, Prelude
Hybrid	High	High	High	High	Midrate	High	Midrate	Watchguard Gamme Firebox, Cisco Intrusion Prevention, McAfee IntruShield

To our best of knowledge, majority of the developed IDPSs could have met IC2, IC5 and IC9 very well. The first characteristic (IC1), is heavily dependent on human supervision. As it was pointed out in challenge number one and two, the traditional IDPSs are not easy to be tailored to new network paradigms and to adapt in new network environments which mean they could not satisfy IC3 and IC4. Although they were successful in recognizing nearly all intrusions (IC6), but they generate unmanageable rate of false alarms which were discussed in challenge number 3. The current developed IDPSs are far from achieving IC7 and IC8 since they are not self-managed systems.

3.2. Alarm management

A common attribute of anomaly IDPSs is that they cannot provide completely accurate detection (Bringas and Penya, 2009). When an IDPS incorrectly identifies benign activity as being malicious, a false positive has occurred. When it fails to identify malicious activity, a false negative has occurred. Unfortunately, the amount of alarms generated by IDPS are unmanageable (thousands of alarms per day) since 99% of them are false positives (Perdisci et al., 2006). This makes alarm investigation both times consuming and error-prone Schubert et al., 2010

With growing anomaly detection application, a new trend on IDPS focused research is shaping which concentrates on how to handle or manage alarms (Elshoush and Osman, 2011). Table 4 lists the most recent research attempting to deal with alarm management problems.

Generally, developed techniques for alert correlation can be categorized in five classifications (Xu and Ning, 2008):

1. Similarity between alert attributes,
2. Predefined attack scenarios,
3. Pre-conditions and post conditions of attacks (constructs the attack scenario by mapping the consequences of earlier attack with pre-requisites of later attack),
4. Multiple information sources (integrates various type of information and may perform reasoning based on alerts and information), and
5. Filtering algorithms.

Majority of researchers have provided a solution to alarm correlation for anomaly techniques (refer to Table 4) since purely anomaly techniques trigger more alarms than other techniques. Although hybrid approach optimizes the visibility and performance of the system, it makes the alarm correlation more complicated. There is a need to attract researchers' attention to provide solutions of alarm management for recently used hybrid detection methods.

4. Intrusion detection and prevention systems in cloud computing

Although distributed IDPSs have been assessed to be capable of protecting securely in large scale networks, but utilization and deployment in cloud computing faces many difficulties and is still a challenging task (Roschke et al., 2009). The variety of cloud

Table 4
Classification of alarm management techniques.

Reference	Year	Method	Performance	Technique category	IDPS technique	Management model
Tjhai et al. (2010)	2010	Two-stage classification system using Self-Organizing Map (SOM) neural networks and k-means algorithm	More than 50% reduction in false Positive rate (FPR)	Similarity between alert attributes and filtering algorithm	Hybrid	Alarm correlation
Mansour et al. (2010)	2010	Data mining technique based on a Growing Hierarchical Self-Organized Map (GHSOM)	Reduces FPR from 15% to 4.7% and false negatives from 16% to 4% for the real-world data	Filtering algorithms and similarity between alert attributes	Anomaly	Alarm correlation
Spathoulas and Katsikas (2010)	2010	Post-processing filter based upon statistical properties of the input alert set	Up to 75% reduction in FPR	Filtering algorithms	Misuse	Alarm quality
Al-Mamory and Zhang (2010)	2010	Filtering using clustering algorithm	Average 82% reduction of FPR	Filtering algorithms	Hybrid	Alarm quality
Li and Tian (2010)	2010	XSWRL ontology technique	No test	Pre-conditions and post conditions of attacks and predefined attack scenarios	Misuse	Alarm correlation
Maggi et al. (2009)	2009	Fuzzy measures and fuzzy sets	Decrease the FPR, but with a small reduction of the detection rate	Similarity between alert attributes	Anomaly	Alarm correlation
Zeng et al. (2009)	2009	Antibody Concentration (NIDMBAC)	FPR was reduced by 8.66%, 4.93% and 6.36% without affecting detection rate	Multiple information sources	Anomaly	Alarm correlation
Vincent Zhou et al. (2009)	2009	Multi- dimensional alert clustering algorithm	Significant reduction in number of alert messages generated	Similarity between alert attributes and filtering algorithm	Anomaly	Alarm correlation
Hoang et al. (2009)	2009	Hybrid fuzzy-based anomaly IDS utilizing hidden Markov model (HMM) detection engine and a normal database detection engine	Reduced FPR by 48%	Similarity between alert attributes	Anomaly	Alarm correlation
Anuar et al. (2008)	2008	Statistical analysis of both attack and normal traffics based on hybrid statistical approach using Data Mining and Decision Tree Classification	Different accuracy results for two models of decision tree and rule-based data mining	Pre-conditions and post conditions of attacks	Unspecified	Alarm correlation
Pietraszek and Tanner (2005)	2005	Data mining (clustering), machine learning (feedback)	More than 50% reduction in FPR	Filtering algorithm	Misuse	Alarm quality

services' users and the complexity of its architecture lead to different requirements and possibilities for being secured by IDPS. In addition to security issues created by its unique features and architecture, cloud computing inherits all the existing systems and networks' security issues (Khorshed et al., 2012).

In order to address the requirements of IDPS for cloud computing, first we look at the special characteristics of cloud computing systems and facing challenges of IDPS development in cloud computing. Then the current developed systems are investigated in terms of their efficiency and effectiveness to deploy on cloud computing environment. Finally, the requirements are developed according to the cloud computing systems' characteristics and the desired characteristic of IDPS.

4.1. Characteristics of cloud computing systems

The identification of the exact characteristics of a target environment is essential to establish system requirements and system development. The characteristics of Cloud computing systems are:

CC1: *Elasticity* is a crucial core feature for cloud systems which confines the underlying infrastructure capability to adapt to changing requirements such as amount and size of data used in an application. Cloud computing involves two types of vertical and horizontal scalability. The vertical scalability refers to the size of the instances and implicit to the amount of resources which are required for maintaining the size. But, horizontal scalability denotes the amount of instances to satisfy changing amounts of requests. This feature in cloud computing comes beyond the other phenomena since it should enable dynamic integration and extraction of physical resources to the infrastructure by two states of up-scaling and down-scaling. This poses extra requirements from middleware management aspect, especially regarding reliability.

CC2: *Reliability* is the capability of ensuring the continuity of the system operation without disruption such as loss of data or code reset during execution. Reliability is normally achieved through utilizing redundant resources, however, majority of the solutions are software-based not hardware- based. There is a strong relationship between availability and reliability – however, reliability focuses in particular on prevention of loss (of data or execution progress).

CC3: *Quality of Service* (*QoS*) support is vitally important for specific requirements which should be met through the provided services or resources. In order to ensure that the accepted service quality of the cloud user in Service Level Agreement (SLA) is met the basic metrics of QoS such as safety, response time and throughput must be guaranteed. Reliability is an aspect of QoS.

CC4: *Agility and adaptability* are two key features of great concern to cloud systems relevant to the elastic capabilities. They refer to on-time reaction to changes in the size of resources and the amount of requests as well as adaptation to changes according to the conditions of environment. This adoption may need different types of resources, different routes or even different qualities. In summary, agility and adaptability require management of the resources to be autonomic.

CC5: *Availability* of services lies in the ability of providing redundant services and data to mask failures transparently. Fault tolerance also needs this ability to introduce new

redundancy such as fresh or previously failed nodes, in an online fashion without or with a little performance penalty. With the increase of simultaneous access, availability is attained through replication of services or data and disseminating them across various resources. This regards to the primary essence of scalability in cloud services and systems.

In summary, it can be concluded that the current systems are not capable to deploy on cloud computing environments which have their own special essence and requirements. There is no traditional IDPS to meet these characteristics efficiently.

4.2. Challenges of IDPS development in cloud computing environments

It is very important to identify the challenges which are originated from cloud computing phenomena before developing an IDPS. The specific challenges that developers face during developing IDPS for cloud computing environments include:

1. In traditional IDPS, due to static essence of the monitored systems, the policies tend to be static since the node groups have stable requirements which have been identified over time. In contrast with traditional mode, the monitored virtual machines are dynamically added and removed. Moreover, the security requirements of each virtual machine tend to be varied (Arshad et al., 2012).
2. The security policies are usually established and managed by a system administrator responsible for security of the whole system. Cloud has several system security administrators; this poses negative effects on intrusion response time. The human intervention would slow down the response time.
3. Engaging to malicious activity of an insider is easily accessible by joining an attacker to a cloud service provider. Meanwhile, the recent researches have provided evidence that most of the intruders come from insiders (Kizza, 2009). Most of the available suggestions to solve this problem are mainly on monitoring employee activities and formulation of cloud providers' policy (Khorshed et al., 2012).
4. The shared infrastructure and virtualization technology put more vulnerability on cloud computing. Any flaw in hypervisors, which allow creating virtual machines and running multiple operating systems, exposes inappropriate access and control to the platform (Grobauer et al., 2011).
5. A very important issue in cloud computing is data transfer cost (Dastjerdi et al., 2009). For example, in Amazon Cloud the data transfer cost is about at $100 to $150 per terabyte. Therefore, new researches should try to provide data cost effective solutions for IDPS in cloud environment with reducing the network bandwidth.
6. Additional issues concern visibility into the inter virtual machine 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 (Viega, 2009). Besides, the new platforms of virtualization themselves would have vulnerabilities that may lead to big compromise, therefore they should be monitored and assessed for configuration errors, patches and so on.
7. 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 (Wang et al., 2009). 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 Cloud-Security-Alliance, 2010. 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 Foster et al., 2008.

Considering the cloud computing characteristics and these mentioned challenges of development of CIDPS, the next subsection reviews the state of the art.

4.3. State of the art of cloud-base IDPS (CIDPS)

Most of the current proposed IDPSs which work on cloud operate at each of the infrastructure, platform, and application layers separately, and mainly support detection and prevention independent from the other layers (Subashini and Kavitha, 2011). For the operating CIDPS in infrastructure layer, Tupakula et al. proposed a model based on a virtual machine monitor, called hypervisor, to protect from different types of attacks in the infrastructure layer (IaaS) (Tupakula et al., 2011). Their model improved the reliability and availability of the system, because the infrastructure can be secured most of the time, and running services can rely on the secure infrastructure. This model has not presented any solution to heal the system if the infrastructure collapsed due to the high severe attacks over the system. A virtual machine monitor solution embeds as a software layer to control the physical resources and it allows running multiple operating systems. The virtual machine monitors are capable of improving the efficiency of attack detection and prevention in CIDPS because they have complete control of the system resources and good visibility of the internal state of the virtual machines. Thus, this solution can overcome the challenge number 1 (see Section 4.2) where monitoring virtual machines are dynamically added or removed.

Majority of researchers have overlooked at the prevention capability in their proposed systems. Gustavo and Miguel (2011) implemented several anomaly-based intrusion detection techniques, and presented an IDS for a reasonably complex Web application designated as SaaS . They found the anomaly-based intrusion detection technique as a promising technique to be used in the application layer, because 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 attacks which may change or inject the false data into the cloud computing system. But they did not suggest any solution for prevention of attacks. Machine learning is the other method which has been used to train the system for anomaly detection. Vieira et al. (2010) proposed a Grid and Cloud Computing Intrusion Detection System (GCCIDS), which covers attacks by using an audit system through integrating misuse and anomaly techniques to detect specific intrusions . The authors used Artificial Neural Network (ANN) to train the system and developed a prototype using a middleware called Grid-M at the University of Santa Catarina, Brazil. They proved that their system had low processing cost while maintaining satisfactory performance for real-time implementation, because it only performed the analysis individually on each node which resulted in lower data transmission traffic from one node to another; and it decreased the complexity of the system. This solution meets CC5 and overcomes challenge number 5 (see Section 4.2) in CIDPS challenges since it performs audit data analysis individually in each node. The drawbacks of GCCIDS are that it only can detect specific intrusions, and does not have the ability of prevention attacks. Although GCCIDS is proposed for both grid and cloud environments but they are different in terms of their security policies, systems requirements and business models (Foster et al.,

2008), which compels specific IDPS design for cloud and grid 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. (2010) developed a collaborative IDS with a central management approach which provided fast and accurate detection . 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 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 techniques using pattern matching of multiple false login attempts and access right violations (Dhage et al., 2011). Their proposed structure suffers from the challenges of lack of scalability and sensitivity of central manager failure. In contrast with this structure, the system developed by Kholidy and Baiardi (2012) has no central manager coordinator. Their fully distributed system with P2P network architecture, hybrid detection techniques using network and host based audit data provides a flexible, robust and elastic solution for cloud computing. 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, leave alone preventions.

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 (Patel et al., 2009). 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 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. (2010). 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, and 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; and 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. (2010) presented a model for malware detection, uCLAVS, based on intrusion ontology representation for cloud computing Web services. Their idea refers to a new concept in IDPS as 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 meanings 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 supported high detection accuracy with least false alarms, but trades-off a lack of program semantics for greater malware resistance and ease of deployment (Azmandian et al., 2011). Using ontology could fill this semantic gap.

Some of the researchers tried to utilize the available resources and optimize the response through *risk assessment and analysis* with a fuzzy logic approach. Lee et al. (2011) 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. It means that logs generated by intruder who has highest anomaly level or security risk are audited with top priority. Therefore, their proposed IDS responses based on the assessed user risks in a way that system does not react against suspicious activities with low risk; this leads to an increase of resources availability. The major drawback is that their IDS is not robust enough to detect large scale (distributed) attacks since each IDS works independently. Takahashi et al. (2010) leveraged ontology and risk assessment approaches and introduced an ontological IDS on cloud computing which works as entity-based and is equipped with a scoring systems for vulnerabilities and weaknesses. 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 meaning in cloud computing.

A virtualization-based NIDPS for cloud computing environment was proposed by Jin et al. (2011). which used network data flow monitoring and real time file integrity Their proposed NIDPS had no control over the host, which increased the vulnerability of insider attacks. In a comparative review on most of the popular commercial NIDPSs in cloud computing conducted by Gunasekaran (2012), Snort was financially, technically and administratively easier to be implemented in small networks. The reason is that rules can be defined at the application layer of a packet in Snort which gives the possibility to collect traffic data, specifically in application layer. They have concluded that Snort is useful when it is not cost efficient to deploy commercial NIDPS. As cost was always a major concern in developing CIDPS, Masud et al. (2008). formulated both of the malicious code detection and botnet traffic detection problems to introduce a new classification ensemble 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 In a research by Dastjerdi et al. (2009) it is proposed to apply mobile agents in IDPS to provide flexible, scalable, and a cost effective system for the cloud environment . However, they believed that this approach does not support enough robustness because of inefficient knowledge sharing between the mobile agents. Beside the available research on CIDPS, Zargar et al. presented a distributed, collaborative, and data driven IDPS (DCDIDPS) which works on three logical layers of network, host, and global as well as platform and application levels. It maximizes the security and detection accuracy, because it monitors every changes and traffic which is gone 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 (Zargar et al., 2011).

Among the reviewed papers, individual IDPS on each node increases the reliability of the system, but it requires higher traffic

data exchanges to synchronize the inter-operative nodes in the cloud environment, as well as result in increased processing time. Beside the structure of IDPS, detection technique is the other major factor that researchers paid a serious attention in their research. Anomaly and hybrid are the most common techniques which are discussed.

Signature based system is faster because it only recognize the limited number of intrusions while anomaly learns the traffic and actions to identify the safe activities and potential intrusions. Machine learning in anomaly consumes more time than knowledge base/database searching in signature based. The models which employed both types are known as hybrid which they have the best accuracy and performance among the other individual methods.

Table 5 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 and system structure. Current conducted researches and proposed solutions are still very far from an ideal IDPS for cloud computing because they not only lack desired IDPS characteristics (ICs), but they also failed to touch cloud systems characteristics thoroughly (CCs). However, they could meet CC5, and CC1, CC2, CC4 partially, but they could not consider all the features which should be addressed in their systems. These inefficiencies are evidence to lack of knowledge of proper requirements which should be identified before initiating any development.

4.4. CIDPS requirements

Taking the characteristics of cloud computing systems and an ideal IDPS into account, the requirements of CIDPS from high-level point of view are identified as follows.

R1: *Handle large-scale dynamic multi-tiered autonomous computing and data processing environments*
Clouds are defined as large scale virtual machine 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 informed from the changes in the resources, but in cloud computing which involve large scale networks and systems it is crucial to maintain these changes automatically without human intervention. To overcome the complexity of its dynamic nature, the CIDPS system should be able to manage itself with least or no human intervention which facilitates the monitoring and control of network elements in real-time. This requirement supports CC2 and CC3.

R2: *Detect variety of attacks with least False Positive Rates*
Due to the growth of attacks, complexity and unpredictability, it is necessary for the system to recognize the new attacks and their vulnerable intention to choose the best response according to the risk severity and proper prevention strategy. The system should be learnable and improve its detection capability over time to support IC6. It also needs to be designed to maintain a desired level of performance and security at the same time with least computation resources since cloud services efficiency relies on its computation capability. Therefore, effective techniques should be utilized to handle false positive alarms maintaining detection performance. It covers CC3.

R3: *Super Fast detection and prevention*
Sharp detection and prevention is a very important enabling factor for CIDPS since it affects the whole system performance and is crucial to deliver the pre-agreed QoS. It refers to CC2 and CC3. A cloud based system with several administrators should have minimized or no human intervention to avoid wasting time for administration responses. It should work in real-time and provides automated responses to suspicious activities. This satisfies IC1 and IC5.

R4: *Self-Adaptive Autonomically*
The feature of easy to adapt to the cloud context in extent which it is supposed to operate 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 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. Design of such system should be flexible enough to be able to cover future developed standards. It supports IC3, IC4, IC8, CC4 and CC5.

R5: *CIDPS Scalability*
A CIDPS should be scalable in order to efficiently handle the massive number of network nodes available in cloud and their communication and computational load. It must scale as the

Table 5
Proposed CIDPSs for cloud computing classified according to our taxonomy.

Reference	Year	Detection technique	Technology layout	Time of detect	Response type	Audit source location	Management structure	Data diffusion	Prevention capability
Vieira et al. (2010)	2009	Hybrid (signature & anomaly)	N/A	Real time	Active	Host & Network	Collaborative	Distributed	No
Dastjerdi et al. (2009)	2010	N/A	Wireless (mobile-agent)	Real time	N/A	Network	Collaborative	Distributed	Yes
Tupakula et al. (2011)	2011	Hybrid (signature & anomaly)	N/A	Real time	Active	Network	Individual	Distributed	Yes
Gustavo and Miguel (2011)	2011	Anomaly	N/A	Real time	Active	Network	N/A	Distributed	No
Smith et al. (2010)	2010	Anomaly	N/A	Real time	Active	N/A	N/A	Distributed	No
Lee et al. (2011)	2011	Anomaly	N/A	Real time	Active	Host & Network	Individual	Distributed	No
Kholidy and Baiardi (2012)	2012	Hybrid (signature & anomaly)	Wireless	Real time	Active	Host & Network	Collaborative	Distributed	No
Dhage et al. (2011)	2011	Anomaly	N/A	Real time	Active	Host	Individual	Distributed	No
Takahashi et al. (2010)	2010	Anomaly	N/A	Real time	Active	Network	Collaborative	Distributed	Yes
Jin et al. (2011)	2011	Anomaly	N/A	Real time	Active	Network	Collaborative	Distributed	Yes

N/A=Not applicable.

nodes added to fit into large cloud dimensions. The placement of detection and correlation handler also affects the scalability and performance of CIDPS. It refers to CC1.

R6: *Deterministic*

Cloud computing provides mission crucial and critical functional services which have specific performance requirements, in terms of latency, reliability and resilience. Supporting IC2, IC9, CC2, CC3, and CC5, a CIDPS should be able to provide and maintain an acceptable level of service in the face of faults, be highly reliable and deliver very high uptime services with imposing minimal overhead. A CIDPS should not only ensure real time performance, but also ensure that the deterministic nature of network is not adversely affected. On the other word, the performance of the monitored systems should not be affected by undue burden of the CIDPS. Further level of network traffic regularity within cloud is subject to change but the performance of the CIDPS should remain deterministic. It is necessary to ensure that the CIDPS has enough capacity to process all the information. Sharing the diagnostic capabilities and the computational load to autonomous agents through the network would increase the level of fault tolerance.

R7: *Synchronization of autonomous CIDPS*

A CIDPS is effectively a massive collaborative IDPS consisting of a large number of autonomous IDPS. While each system operates and detects intrusions and anomalies independently, their information and activities must be synchronized in order to recognize distributed and concurrent attacks, apply appropriate response or modify a particular component system or the whole network configuration, and adopt proper prevention strategy.

R8: *Resistance to Compromise*

Referring to IC7, a CIDPS must protect it-self from unauthorized access or attacks. A CIDPS should be able of authenticating network devices and IDPSs mutually, authenticating the administrator and auditing his actions, protecting its data, and blocking any loopholes which may create additional vulnerabilities.

All new solutions for development of a CIDPS should consider the above requirements to be able to overcome cloud computing complexities and met the real operational goals of the cloud computing world. As shown in Table 6, from our analysis the proposed CIDPSs given in the references do not meet all the requirements and are not realistic to be placed on actual cloud computing environments. The heterogeneous essence of cloud computing necessitate using hybrid solutions and hybrid techniques for CIDPS to meet the stated requirements. From these set of requirements the criteria to judge the capabilities of CIDPS can be easily formulated.

Table 6
The developed CIDPSs which met our proposed requirements.

References requirements	*R1*	*R2*	*R3*	*R4*	*R5*	*R6*	*R7*	*R8*
Vieira et al. (2010)	×	√	×	√	√	√	×	√
Dastjerdi et al. (2009)	×	***N/A***	***N/A***	√	√	√	×	***N/A***
Tupakula et al. (2011)	×	***P***	×	√	√	×	***N/A***	√
Gustavo and Miguel (2011)	×	***P***	×	***N/A***	***N/A***	×	×	√
Smith et al. (2010)	×	√	***P***	***N/A***	***N/A***	***P***	***N/A***	***P***
Lee et al. (2011)	×	√	×	***N/A***	***N/A***	√	***N/A***	×
Kholidy and Baiardi (2012)	***P***	√	×	√	√	√	×	√
Dhage et al. (2011)	×	√	***N/A***	√	√	×	***N/A***	√
Takahashi et al. (2010)	√	√	√	√	***N/A***	***P***	×	√
Jin et al. (2011)	√	***P***	×	√	√	***P***	***N/A***	***N/A***

P=Partially *X*=Does not meet requirement √=meet requirement N/A=Not applicable.

5. Discussion

To respond to the first research question (*what criteria and requirements should an IDPS meet to be deployed on cloud computing environments*?) a list of requirements was gathered (in Sub-Section 4.4) based on the characteristics of cloud computing systems and IDPSs.

In this Section, the possible solutions that meet the list of CIDPS requirements are discussed to find the answer of the second research question (*which methods or techniques can satisfy these requirements*?). Due to the complexity of CIDPS, we grab four concepts of *Autonomic Computing*, *Risk Management*, *Fuzzy Theory*, and *Ontology* as shown in Fig. 4 from our state of the art review of CIDPS to satisfy the requirements (which were also shown in Fig. 1 in the *red box* as "Advanced Components of IDPS").

Let's see how these four concepts can help to design an efficient system which meets the requirements of the CIDPS. In ***R1***, it was discussed that the system should be self-managed to handle a dynamic environment. The self-configuring characteristic enables the system to detect hardware and software changes automatically and seamlessly. With ontology knowledge base, intrusion sensors can react and respond dynamically to changing networks and threats as well as leverage integral data from other sources on the network. Because ontology allows defining concepts, objects, and relationships in a knowledge domain to unify the knowledge base of the system; this unified knowledge base facilitates providing reasoning framework, intelligence, and inference.

The ***R2*** acknowledged the need to detect various attacks with least FPR. Using the hybrid detection techniques and appropriate risk management and severity analysis approaches can satisfy this requirement. Once a threat is determined, the system should scan impacted systems and go deeper into the vulnerability detected. The data of vulnerability assessment can then be analyzed in correlation with network behavioral data; it will make a true real-time picture of which attacks are occurring and help to assess its possible impact on the target system. Once a criticality rating has been assigned to assets and a continuous stream of ontology was gathered, then intrusion prevention solutions can begin to take proactive actions dynamically to reduce operational overhead. For example, intrusion prevention rules which are not applicable to certain systems and applications in a specific IP range can be disabled, this reduces false positives significantly. These rules may be re-enabled if new data certifies that a particular system has become vulnerable to a known

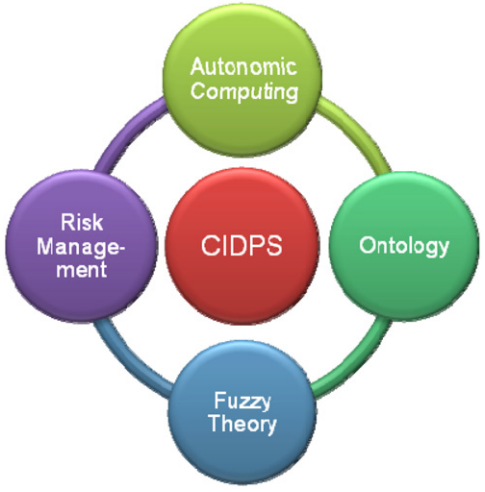


Fig. 4. Proposed solution for the use of techniques of these concepts to develop a CIDPS.

attack. For a real dynamic response, sensors can distribute short-term modifications to block malicious traffic immediately. This real-time protection and prevention push the system to a state of continuous monitoring, assessment and optimizing. To properly analyze false alarm reduction strategy, it is necessary to quantify the risk exposed to the attacked assets and the residual risk conveyed by the asset.

However, risks and intrusions have different consequences and dangers which should be considered. Although, the system should prevent and detect all types of intrusions and attacks, but it is necessary to identify the danger level and intensity of the risk. In some asynchronous attacks and lack of enough resources to prevent all system penetrations, CIDPS can make priority based on dangers level to respond properly and result in the least vulnerable and possible infection. Fuzzy logic can also help to score vulnerable assets, determine likelihood levels for threats, assess the associated relative risk, prioritize the alarms and plan a proper strategy for response. They all can be characterized by domain ontology including high-level concepts (such as attacks, vulnerabilities and incidents) to improve the use of obtained knowledge.

As discussed earlier, intrusions are evaluated and scored in different levels from different aspects, so they can be mapped on Multi-Dimensional Type-2 fuzzy logic (Castillo, 2012). In Multi-Dimensional Type-2 the logic is as the same as classic fuzzy logic, however, the fuzziness and fuzzification steps consider every aspects of intrusion at the same time as a whole not separately.

Speed is a key element rectified in ***R3***. An automated agent-based and self-managed mechanism can reduce the response time significantly by eliminating the elapsed time from alert generation till system administrator response. In case of any corruption, the self-healing property comes to help the system to correct itself by identifying the errors, diagnosing the problem and processing rerun without human intervention.

However, ***R4*** is concern with structure and architecture of CIDPS, but a self-optimized system can also facilitate adoptability by optimizing its use of resources and communicating with other systems to transfer the data and files. The CIDPSs become more adaptive and real-time by using the same ontologies which facilitates communicating and knowledge sharing.

Scalability and handling the large number of network nodes is the major concern of ***R5***. In a cloud environment with a very large network and heavy traffic, CIDPS is challenged with more difficulty to see all traffic on a switched network. This problem has shaped a new approach with looking closer at the end-point host connecting to the network access-point (this trend is also observable in Table 1). However, the most effective deployment is to combine both of the host-based and network-based, while few vendors were able to offer this (Beale et al., 2004).

Risk management techniques and autonomic computing with all self-managing properties can satisfy ***R6***. Autonomic computing can bring the same performance to CIDPS as a Human Nervous System. The nervous system controls our unconscious reflexes without us being aware of this, and can provide fault-tolerance in the system. It can keep its functional continuity even when its sensors fail.

Using the ontologies and mobile agents can help to synch and transfer messages between CIDPSs as it is the target aim of ***R7***. Mobile agents are assumed to have incomplete information since they operate in complex, dynamic, and non-deterministic environment of cloud computing without a global control to synchronize the data. Thus, communication plays a significant role for agents to share the information, synch or co-ordinate their actions, and manage the interdependencies. Intelligent interoperability between the mobile agents can be achieved by using common ontologies and interpretative knowledge allowing agents to cooperate while maintaining their autonomy. These agents can exchange their knowledge which shares the same ontology. Mobile agents can benefit from virtualization platform that cloud computing provides because virtual machines are ideal for agents to execute their program safely. The usability of a virtual machine to provide secure, isolated sand boxes for the mobile agents is acknowledged by Topaloglu and Bayrak (2008). As the previous section reviewed, the major issue of using mobile agents is inefficient knowledge sharing between mobile agents. Employing ontology can fill this gap since it provides agents with a common interpretation of the environment. Distributed and collaborative structure of intrusion detection and prevention within cloud systems help to decrease the complexity of redundant monitoring of attack flows at different check points.

The self-protecting property of autonomic computing can anticipate detection and protection of the system itself against threats as is concern of ***R8***. A CIDPS equipped with this property is able to detect security incidents while they occur and take proper response and corrective actions to make them less vulnerable. Furthermore, using the autonomous agents mitigates the risk of compromising the system since it is difficult for a single attack to affect all the agents in the system due to the heterogeneous essence of the agents.

Finally, it is worthy to note that there should be a balance between system security level and system performance due to their trade-off relationship. An IDPS that provides highly secured and trustworthy services uses more patterns and rules. Therefore, it needs more computing resources for supplying better security. Extending this situation to cloud computing, the allocated resources to cloud costumers will decrease (Lee et al., 2011). So, the best solution is not necessarily a very complex system using many resources and rules, but is an optimized design and using smart techniques which make the system independent by self-managing and self-learning.

6. Conclusion

This paper presented a comprehensive taxonomy and state of the art of intrusion detection and prevention systems to drag researchers' attention for possible solutions to intrusion detection and prevention in cloud computing. A specific attention was given to cloud systems characteristics and current challenges banning IDPS development for cloud. A list of requirements for a cloud based intrusion detection and prevention system was provided along with grabbing four applicable concepts in developing a CIDPS from our review on latest researches: autonomic computing, ontology, risk management, and fuzzy theory for making an ideal design to meet these requirements. In the future, we plan to develop a fully fledged framework and design a CIDPS by leveraging these concepts.

Acknowledgement

The authors wish to 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

Al-Mamory S, Zhang H. New data mining technique to enhance IDS alarms quality. Journal in Computer Virology 2010;6:43–55.

Anuar NB, Sallehudin H, Gan A, Zakari O. Identifying false alarm for network intrusion detection system using data mining and decision tree. Malaysian Journal of Computer Science 2008;21:101–15.

Arshad J, Townend P, Xu J. A novel intrusion severity analysis approach for Clouds. Future Generation Computer Systems 2012.

Awodele O, Idowu S, Anjorin O, Joshua VJ. A multi-layered approach to the design of intelligent intrusion detection and prevention system (IIDPS). Issues in Informing Science and Information Technology 2009;6.

Axelsson S. Intrusion Detection Systems: A Taxonomy and Survey. Technical Report No 99–15," Department of Computer Engineering.Sweden: Chalmers University of Technology; 2000.

Azmandian F, Moffie M, Alshawabkeh M, Dy J, Aslam J, Kaeli D. Virtual machine monitor-based lightweight intrusion detection. SIGOPS—Operating Systems Review 2011;45:38–53.

Beale J, AR Baker,B Caswell, and M Poor, "Snort 2.1 Intrusion Detection," ed.: Syngress Media Inc, 2004, p. 25.

Bringas PG, Penya YK. Next-generation misuse and anomaly prevention system. Enterprise Information Systems 2009;19:117–29.

Bringas PG, Penya YK. In: Filipe J, Cordeiro J, editors. Next-Generation Misuse and Anomaly Prevention System Enterprise Information Systems, 19. Berlin Heidelberg: Springer; 2009. p. 117–29.

Byrski A, Carvalho M. In: Bubak M, van Albada G, Dongarra J, Sloot P, editors. Agent-Based Immunological Intrusion Detection System for Mobile Ad-Hoc Networks Computational Science—ICCS 2008, 5103. Berlin/Heidelberg: Springer; 2008. p. 584–93.

Carl G, Kesidis G, Brooks RR, Rai S. Denial-of-service attack-detection techniques. Internet Computing, IEEE, 2006;10:82–9.

Castillo O. Type-2 Fuzzy Logic in Intelligent Control Applications, in Type-2 Fuzzy Logic.Berlin/ Heidelberg: Springer; 2012 pp. 7–22..

Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Computing Surveys 2009;41:1–58.

Chen M, Kwon T, Yuan Y, Leung V. Mobile agent based wireless sensor networks. Journal of Computers 2006;1:14–21.

Choo K-KR. The cyber threat landscape: challenges and future research directions. Computers & Security 2011;30:719–31.

Chung-Ming O. Host-based intrusion detection systems adapted from agent-based artificial immune systems. Neurocomputing 2012.

Cloud-Security-Alliance. (2010). Top Threats to Cloud Computing V1.0. Available: 〈https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf〉.

Dastjerdi AV, KA Bakar, and SGH Tabatabaei, "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.

Dhage S, B Meshram, R Rawat, S Padawe, M Paingaokar, and A Misra, "Intrusion detection system in cloud computing environment," in *International Conference & Workshop on Emerging Trends in Technology*, New York, NY, USA pp. 235–9, 2011.

Elshoush HT, Osman IM. Alert correlation in collaborative intelligent intrusion detection systems—a survey. Applied Soft Computing 2011;11:4349–65.

Estevez-Tapiador JM, Garcia-Teodoro P, Diaz-Verdejo JE. Anomaly detection methods in wired networks: a survey and taxonomy. Computer Communications 2004;27:1569–84.

Foster I, Y Zhao, I Raicu, and S Lu, "Cloud computing and grid computing 360-degree compared," in Grid Computing Environments Workshop, 2008. GCE '08 Austin, TX. pp. 1–10, 2008.

Gaffney JEJr and JW Ulvila, "Evaluation of intrusion detectors: a decision theory approach," in *IEEE Symposium on Security and Privacy*, 2001. *S&P 2001*, Oakland, CA, USA. pp. 50–61, 2001.

Galante J., O Kharif, and P Alpeyev (2011, May 17, 2011). *Sony Network Breach Shows Amazon Cloud's Appeal for Hackers*. Available: 〈http://www.bloomberg.com/news/2011-05-15/sony-attack-shows-amazon-s-cloud-service-lures-hackers-at-pennies-an-hour.html〉.

García-Teodoro P, Díaz-Verdejo J, Maciá-Fernández G, Vázquez E. Anomaly-based network intrusion detection: techniques, systems and challenges. Computers & Security 2009;28:18–28.

Grobauer B, Walloschek T, Stocker E. Understanding cloud computing vulnerabilities. Security & Privacy, IEEE 2011;9:50–7.

Gunasekaran S., "Comparison of network intrusion detection systems in cloud computing environment," in *international conference on computer communication and informatics (ICCCI)*, Coimbatore, pp. 1–6, 2012.

Gustavo N, Miguel C. Anomaly-based intrusion detection in software as a service. Dependable Systems and Networks Workshops 2011:19–24.

Herrero Á, Corchado E. In: Abraham A, Hassanien A-E, de Carvalho A, editors. Mining Network Traffic Data for Attacks through MOVICAB-IDS Foundations of Computational Intelligence, 4 204. Berlin/ Heidelberg: Springer; 2009. p. 377–94.

Hoang XD, Hu J, Bertok P. A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference. Journal of Network and Computer Applications 2009;32:1219–28.

Hwang K, Cai M, Chen Y, Qin M. Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. Dependable and Secure Computing, IEEE Transactions on 2007;4:41–55.

Jaiswal A, Jain S. Database intrusion prevention cum detection system with appropriate response. International Journal of Information Technology 2010;2:651–6.

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 2011:1–19.

Kazienko P, Dorosz P. Intrusion Detection Systems. Windowsecurity 2004.

Kerschbaum F, Spafford EH, Zamboni D. Using internal sensors and embedded detectors for intrusion detection. Journal of Computer Security 2002;10:23–70.

Klüft S, Alarm management for intrusion detection systems—prioritizing and presenting alarms from intrusion detection systems," Master, Computer Science Programme, master of science thesis, University of Gothenburg, 〈http://hdl.handle.net/2077/28856〉, 2012.

Khanum S, Usman M, Alwabel A. Mobile agent based hierarchical intrusion detection system in wireless sensor networks. International Journal of Computer Science Issues, IJCSI 2012;9.

Kholidy HA and F. Baiardi, CIDS: a Framework for Intrusion Detection in Cloud Systems," in *Ninth International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, NV, pp. 379–5, 2012.

Khorshed MT, Ali ABMS, Wasimi SA. A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. Future Generation Computer Systems 2012;28:833–51.

Kizza JM. System intrusion detection and prevention. In: Kizza JM, editor. A Guide to Computer Network Security. London: Springer; 2009. p. 273–98.

Lee JH, MW Park, JH Eom, and TM Chung, "Multi-level Intrusion Detection System and log management in Cloud Computing," in *13th international conference on advanced communication technology (ICACT)*, Seoul, pp. 552–5, 2011.

Lee JH, MW Park, JH Eom, and TM Chung, "Multi-level intrusion detection system and log management in cloud computing," 13th international conference on advanced communication technology (ICACT), pp. 552–5, 2011.

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.

Liang G, Li T, Ni J, Jiang Y, Yang J, Gong X. An immunity-based dynamic multilayer intrusion detection system. In: Huang D-S, Li K, Irwin G, editors. Computational Intelligence and Bioinformatics, 4115. Berlin Heidelberg: Springer; 2006. p. 641–50.

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, editors. Recent Advances in Intrusion Detection, 2516. Berlin/Heidelberg: Springer; 2002. p. 307–26.

Li Y, Jing C, Xu J. In: Li K, Fei M, Jia L, Irwin G, editors. A New Distributed Intrusion Detection Method Based on Immune Mobile Agent Life System Modeling and Intelligent Computing, 6328. Berlin/Heidelberg: Springer; 2010. p. 233–43.

Li W, Tian S. An ontology-based intrusion alerts correlation system. Expert Systems with Applications 2010;37:7138–46.

Maggi F, Matteucci M, Zanero S. Reducing false positives in anomaly detectors through fuzzy alert aggregation. Information Fusion 2009;10:300–11.

Mansour N, Chehab M, Faour A. Filtering intrusion detection alarms. Cluster Computing 2010;13:19–29.

Martínez CA , Echeverri GI, and Sanz AGC, "Malware detection based on cloud computing integrating intrusion ontology representation," in *IEEE Latin-American Conference on Communications (LATINCOM)*, Bogota, pp. 1–6, 2010.

Masud MM, Al-Khateeb TM, Hamlen KW, Gao J, Khan L, Han J, Thuraisingham B. Cloud-based malware detection for evolving data streams. ACM Transactions Management Information Systems 2008;2:1–27.

Moore AP, Cappelli DM, Trzeciak RF. The "Big Picture" of insider it sabotage across U.S. critical infrastructures. In: Stolfo S J, Bellovin SM, Keromytis AD, Hershkop S, Smith SW, Sinclair S, editors. Insider Attack and Cyber Security, 39. US: Springer; 2008. p. 17–52.

Nazer GM, Selvakumar AAL. Current intrusion detection techniques in information technology—a detailed analysis. European Journal of Scientific Research 2011;65:611–24.

Patcha A, Park J-M. An overview of anomaly detection techniques: existing solutions and latest technological trends. Computer Networks 2007;51:3448–70.

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. 223–24, 2009.

Patel A, Qassim Q, Wills C. A survey of intrusion detection and prevention systems. Information Management and Computer Security 2010;18:277–90.

Perdisci R, Giacinto G, Roli F. Alarm clustering for intrusion detection systems in computer networks. Engineering Applications of Artificial Intelligence 2006;19:429–38.

Pietraszek T, Tanner A. Data mining and machine learning—towards reducing false positives in intrusion detection. Information Security Technical Report 2005;10:169–83.

Rasoulifard A, Ghaemi Bafghi A, Kahani M. Incremental hybrid intrusion detection using ensemble of weak classifiers. In: Sarbazi-Azad H, Parhami B, Miremadi S-G, Hessabi S, editors. Advances in Computer Science and Engineering, 6. Berlin Heidelberg: Springer; 2009. p. 577–84.

Roschke S, F Cheng, and C Meinel, "Intrusion detection in the Cloud," presented at the Eighth IEEE international conference on dependable, autonomic and secure computing, pp. 729–34, 2009.

Scarfone K, Mell P. Guide to Intrusion Detection and Prevention Systems (idps). Special Publication, 800. NIST; 2007 p. 94.

Schubert L, Jeffery K, Neidecker-Lutz B. The future for cloud computing: opportunities for european cloud computing beyond. Expert Group Report, Public Version1, European Commission 2010 2010.

Shabtai A, Fledel Y, Kanonov U, Elovici Y, Dolev S, Glezer C. Google android: a comprehensive security assessment. Secur. Privacy IEEE 2010;8:35–44.

Sharma T, Sinha K. Intrusion detection systems technology. International Journal of Engineering and Advanced Technology (IJEAT) 2011;1:28–33.
Shon T, Kovah X, Moon J. Applying genetic algorithm for classifying anomalous TCP/IP packets. Neurocomputing 2006;69:2429–33.
Shyu M-L, Sainani V. A multiagent-based intrusion detection system with the support of multi-class supervised classification. In: Cao L, editor. Data Mining and Multi-agent Integration. US: Springer; 2009. p. 127–42.
Smith A, Johnson N. A smart sensor to detect the falls of the elderly. Pervasive Computing, IEEE 2004;3:42–7.
Smith D, Q Guan, and S Fu, "An Anomaly Detection Framework for Autonomic Management of Compute Cloud Systems," *34th Annual Computer Software and Applications Conference Workshops (COMPSACW), Seoul,* pp. 376–1, 2010.
Spathoulas GP, Katsikas SK. Reducing false positives in intrusion detection systems. Computers & Security 2010;29:35–44.
Sproull T, Lockwood J. Distributed instrusion prevention in active and extensible networks active networks. In: Minden G, Calvert K, Solarski M, Yamamoto M, editors. Lecture Notes in Computer Science, 3912. Berlin/Heidelberg: Springer; 2007. p. 54–65.
Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications 2011;34:1–11.
Su M-Y, Yu G-J, Lin C-Y. A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach. Computers & Security 2009;28:301–9.
Sy BK. Integrating intrusion alert information to aid forensic explanation: an analytical intrusion detection framework for distributive IDS. Inf. Fusion 2009;10:325–41.
Takahashi T, Y Kadobayashi, and H Fujiwara, "Ontological approach toward cybersecurity in cloud computing," presented at the Proceedings of the 3rd international conference on Security of information and networks, Taganrog, Rostov-on-Don, Russian Federation, 2010.
Thatte G, Mitra U, Heidemann J. Parametric methods for anomaly detection in aggregate traffic,. IEEE/ACM Transactions on Networking (TON) 2011;19: 512–25.
Tjhai GC, Furnell SM, Papadaki M, Clarke NL. A preliminary two-stage alarm correlation and filtering system using SOM neural network and *K*-means algorithm. Computers & Security 2010;29:712–23.
Topaloglu U, Bayrak C. Secure mobile agent execution in virtual environment. Autonomous Agents and Multi-Agent Systems 2008;16:1–12.
Tupakula U, V Varadharajan, and N Akku, "Intrusion Detection Techniques for Infrastructure as a Service Cloud," *IEEE International Conference on Dependable, Autonomic and Secure Computing* pp. 744–1, 2011.
Viega J. Cloud computing and the common man. Computer 2009;42:106–8.
Vieira K, Schulter A, Westphall C. Ntrusion detection for grid and cloud computing. IT Professional 2010;12:38–43.
Vincent Zhou C, Leckie C, Karunasekera S. Decentralized multi-dimensional alert correlation for collaborative intrusion detection. Journal of Network and Computer Applications 2009;32:1106–23.
Wang C, Q Wang, K Ren, and W Lou, "Ensuring data storage security in cloud computing," in 17th International Workshop on Quality of Service, 2009. IWQoS, Charleston, SC. pp. 1–9, 2009.
Whitman ME, Mattord HJ. Principles of Information Security, ed.: Course Technology Ptr 2011:315.
Wu SX, Banzhaf W. The use of computational intelligence in intrusion detection systems: a review. Applied Soft Computing 2010;10:1–35.
Xie M, Han S, Tian B, Parvin S. Anomaly detection in wireless sensor networks: a survey. Journal of Network and Computer Applications 2011;34:1302–25.
Xin W, H Ting-lei, and L Xiao-yu, Research on the Intrusion detection mechanism based on cloud computing," in 2010 I*nternational Conference on Intelligent Computing and Integrated Systems (ICISS)*, Guilin, pp. 125–8, 2010.
Xu D, Ning P. Correlation analysis of intrusion alerts.Intrusion Detection Systems, 38. US: Springer; 2008 65–92.
Yue WT, Çakanyıldırım M. A cost-based analysis of intrusion detection system configuration under active or passive response. Decision Support System 2010;50:21–31.
Zargar ST, H Takabi, and JBD Joshi, "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*, 2011.
Zeng J, Li T, Li G, Li H. A new intrusion detection method based on antibody concentration emerging intelligent computing technology and applications. In: Huang D-S, Jo K-H, Lee H-H, Kang H-J, Bevilacqua V, editors. With Aspects of Artificial Intelligence, 5755. Berlin/ Heidelberg: Springer; 2009. p. 500–9.
Zhou CV, Leckie C, Karunasekera S. A survey of coordinated attacks and collaborative intrusion detection. Computers & Security 2010;29:124–40.