Proposed Architecture for Intrusion Detection System for Software as a Service in Cloud Computing Environment

Azuan Ahmed[1], Ganthan Narayana Samy[1], Bharanidharan Shanmugam[2], Norbik Bashah Idris[1] and Suriayati Chuprat[1]

[1]*Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia*

[2]*Charles Darwin University, Ellengowan Drive, Casuarina Campus, Australia*

azuan2@live.utm.my, ganthan.kl@utm.my, Bharanidharan.Shanmugam@cdu.edu.au, norbik@utm.my, suria.kl@utm.my

Abstract

The purpose of this paper is to propose an architecture for intrusion detection based on Software as a Service (SaaS) called Software as a Service Intrusion Detection Services (SaaSIDS) in a cloud environment. Therefore, this research focusing on developing Software As A Service IDS (SaaSIDS) where the traffic at different points of the network is sniffed and the interested packets would be transferred to the SaaSIDS for further inspection. The main engine of SaaSIDS is the hybrid analysis engine where the signature based engine and anomaly based engine which using artificial immune system will work in parallel. The SaaSIDS is able to identify malicious activity and would generate appropriate alerts and notification accordingly.

Keywords: *Intrusion Detection System (IDS); software as a service (SaaS); software as a service intrusion detection services (SaaSIDS); cloud computing*

1.0 INTRODUCTION

Cloud Computing is a new implementation of computer technology and open a new research area and create a lot of opportunity of exploration. One of the new implementation in cloud is adopting Intrusion Detection System (IDS). Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction (Mell and Grance 2011). Cloud computing is based on five attributes namely on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. (Mell and Grance 2011). On demand self-service refers to such as server time and network storage, as needed automatically without requiring human interaction with each service provider. Broad network access refers to capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms. Resource pooling is computing resources are pooled to serve multiple consumers using a multi-tenant model. Rapid elasticity is a capability can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. Measured service refers to cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (Mell and Grance 2011).

There are three fundamental service models that are being implemented by cloud service provider namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) that will be explain further detail in literature review section (Farhan B.S. and Sajjad H. 2011.; Mark T., et al., 2011). There's a problem with the implementation of IDS in normal environment. Traditional IDS need a lot of self maintenance and did not scale with the customer security requirements. The cost of maintaining and installing the traditional IDS is also a big consideration in

implementing IDS in an organization. One of the solutions of the problems in traditional IDS is by implementing it in a cloud environment. This paper is organized into four sections. The next section describes the literature review related to this research. Section 3 explains a proposed architecture in this research and followed by conclusions in section 4.

2.0 LITERATURE REVIEW

Intrusion Detection System (IDS) is defined as any software or hardware that monitors network or system for any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource (Shanmugam and Idris, 2009). Traditional IDS need a lot of self maintenance and did not scale with the customer security requirements. In addition, maintenance of traditional IDS requires expertise and consumes more time that normal company did not have (Cymtec Systems, Inc., 2012; Yassin, W., et al., 2012). The cost of maintaining and installing the traditional IDS is also a big consideration in implementing IDS in an organization. In addition, a decentralized traditional IDS approach where being implemented in traditional IDS can increase the network vulnerabilities in the protected system when the IDS system is deployed and implemented together in the same network and made visible to others. The IDS itself are exposed to the internal attacks where attacker from the same network will have access to the IDS and launching attack directly towards the IDS. The IDS must be isolated and invisible from the same network where the host and servers reside (Yassin, W., et al., 2012).

In order to protect computing infrastructures which contains valuable assets from cyber attacks, most enterprises set their strategy to deploy their IDS on dedicated hardware. However, such strategy is no longer effective today when small and medium enterprises (SMEs) are conveniently tapping into the cloud environment which provides them the platform, infrastructure and software as services on a pay-per-use basis (Subashini and Kavitha, 2011). Moreover, IDS is commonly deployed in the traditional way, such as on virtual machines (VM), which is considered more vulnerable with diverse security requirements. In the traditional deployment, the benefits of customization and on-demand operations offered by cloud are contradicted by the lengthy intrusion response time and thus affecting the overall security of the system (Cymtec Systems, Inc., 2012).

2.1 Artificial Immune System (AIS)

Kim, J., et al. (2007) stated since 1993, researcher start to implement Artificial Immune System (AIS) to the IDS detection mechanism since AIS can be considered as an anomaly detection with minimal false negative and positive. According to Kephart et al (1994); Somayaji et al (1997); Forrest, S., et al., 1994) was among the first that introduce computational intelligent inspired by AIS in IDS and this idea are still expanding among researchers. Kim, J., et al. (2007) classified AIS implementation to IDS into three major roots:-

2.1.1 Conventional Algorithms

Conventional algorithm introduced by Kephart et al (1994) was among the earliest attempt to apply HIS in IDS. Their research was more on automatic detection of computer viruses and worms because computer interconnectivity becoming more complex and traditional virus detection method (signature-based detection) will become less effective. Their aim was to create a virus detection system that detect and responds to virus or worms automatically. They proposed a system using either of the fuzzy matching algorithms from a signature of viruses or using integrity monitors that monitor important binaries and data in the host for any changes. What makes their system unique is, to reduce false positive, if a binary was suspected as a virus, a decoy (a binary that created for being infected) will be exposed to the suspect and if the decoy are being infected, then they can confirm that that was a virus.

2.1.2 Negative Selection

One of the three major roots of AIS was negative selection (NS). This technique implements the negative selection in the T-cell maturation process (Forrest, S., et al., 1994). In negative selection, the process is eliminating any immature Tcells that bind to self antigens. This will make HIS to detect

non-self antigens without mistake. So, any antigens with T-cells will automatically detect as non-self. As proposed by Forrest, S., et al., (1994), there is three phases of negative selection: defining self, generating detectors and monitor the anomalies. When defining self phase, it is the same process in normal anomaly detection where the system identified the normal behavior patterns. The next phase is generating detector where it generates a number of random patterns that will be compared to each self-pattern defined in the first phase. If any randomly generated pattern matches a self-pattern, this pattern fails to become a detector and thus it is removed. Otherwise, it becomes a detector pattern and monitors subsequent profiled patterns of the monitored system. In the last phase, the detector pattern will be match with any newly profile pattern and if the pattern did not match, then it was detected as anomaly.

2.1.3 Danger Theory

In danger theory, immune response is triggered by unusual death of self-cells. Burgess, M. (1998) proposed that an autonomous and distributed feedback and healing mechanism, triggered when a small amount of damage could be detected at an initial attack. There are still more AIS algorithm to be explored through this research especially that related to genetic algorithm where can be improved to be applied in IDS detection algorithm and be implemented into the cloud environment.

2.1.4 Cloud-based IDS

There are three fundamental service models that are being implemented by cloud service provider namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (Subashini and Kavitha, 2011).

In the most basic cloud service model, providers of Infrastructure as a Service (IaaS) offer computers in physical or virtual machines and other resources. IaaS clouds often offer additional resources such as images in a virtual-machine image-library, raw and file-based storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS cloud providers supply these resources on-demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds which are dedicated virtual private networks. To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis and cost reflects the amount of resources allocated and consumed.

In the Platform as a Service (PaaS) model, cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some PaaS offers, the underlying computer and storage resources scale automatically to match application demand such that cloud user does not have to allocate resources manually.

In the Software as a Service (SaaS) model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. The cloud users do not manage the cloud infrastructure and platform on which the application is running. This eliminates the need to install and run the application on the cloud user's own computers simplifying maintenance and support. What makes a cloud application different from other applications is its scalability. This can be achieved by cloning tasks onto multiple virtual machines at run-time to meet the changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user who sees only a single access point. To accommodate a large number of cloud users, cloud applications can be multitenant, that is, any machine serves more than one cloud user organization. It is common to refer to special types of cloud based application software with a similar naming convention: desktop as a service, business process as a service, test environment as a service, communication as a service.

The implementation of IDS into Cloud environment especially in Software as a Service (SaaS) is a very new approach in IDS. Previous work from W. Yassin et al (2012) proposed a cloud-based IDS Framework where the deployment of IDS is applied into cloud environment. The proposed model is using signature-based detection as the analysis engine. Data from user cloud will be sent to Cloud IDS to be analysed. The problem with the proposed model are the model will not detecting any new attacks that trying to intrude user cloud and the information sent from user cloud to the Cloud IDS are uncompress data that will make traffic congestion at the users cloud network.

3.0 PROPOSED ARCHITECTURE

The proposed architecture as shown in Fig.1. As described in Fig.1, SaaSIDS consist of SaaSIDS sensor, SaaSIDS Service Component and Hybrid Analysis Engine that will be discussed in the following section.

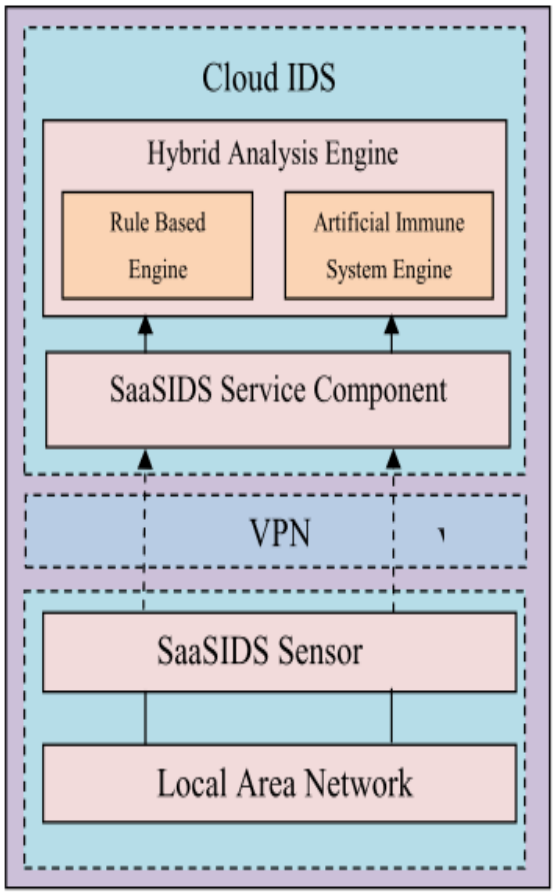

Fig.1 SaaSIDS Proposed Architecture

3.1 DESCRIPTION OF PROPOSED ARCHITECTURE

The aim of this research is to produce a prototype of SaasIDS and the prototype will be tested with the dataset for evaluations. Traffic at different points of the network is sniffed and the interested packets would be transferred to the SaaSIDS for further inspection. The main engine of SaaSIDS is the hybrid analysis engine where the signature based engine and anomaly based engine which using artificial immune system will work in parallel as shown in Fig.1. The SaaSIDS is able to identify malicious activity and would generate appropriate alerts and notification accordingly. We believe the proposed approach offers new opportunities namely providing economic, scalable and viable option to any cloud-based users and satisfy the users' security demands.

3.1.1 SaaSIDS Sensor

SaaSIDS sensor is a device that installed on user's network to collect the selected packet before sending it towards Cloud IDS. SaaSIDS sensor also responsible for compressing and encrypting the packets to reduce the overhead during sending multiple packets into the Cloud IDS. This device will be running on client side which it will monitor all the traffic flowing from and into the client and sending suspected packet to the SaaSIDS Service Component for further analysis.

3.1.2 SaaSIDS Service Component

SaaSIDS Service Component is responsible for analyzing and validating the received information from SaaSIDS Sensor before determining whether to drop the packet or to forward it to the Hybrid Analysis Engine. SaaSIDS Service Component also responsible to decrypt and decompress the information before being processed by the Hybrid Analysis Engine. When the packet received from the SaaSIDS sensor, the packets will first being decrypted and decompressed. Then the SaaSIDS Service Component will forward the packet to the Hybrid Analysis Engine for analysis.

3.1.2 Hybrid Analysis Engine

Hybrid Analysis Engine is the core component of the SaaSIDS. This component consists of two methods of analysis which are Rule Based Engine and Artificial Immune System Engine. Rule Based Engine will analyze the information received for intrusion detection based on the signature and if the information is not detected, Artificial Immune System Engine will analyze the packet by using anomaly based detection. When the packet was received by SaaSIDS Service Component, Hybrid Analysis Engine will start to analyze the packet based on the Artificial Immune System (AIS) engine and Rule Based Engine as stated before.

4.0 CONCLUSIONS

Basically, in cloud, IDS can be managed centrally and can reduce the maintenance need to be done by a single company that using the IDS. Therefore, the future of IDS should come with reasonable cost, and reduced complexity with strong defensive mechanism. Therefore, this research proposes an intrusion detection based on Software as a Service (SaaS) called Software as a Service Intrusion Detection Service (SaaSIDS). Thus, the proposed architecture offers new opportunities namely providing economical, scalable and viable option to any cloud-based users and satisfies the users' security demands.

5.0 ACKNOWLEDGEMENTS

The authors would like to thank Universiti Teknologi Malaysia (UTM) for supporting this work through Research University Grant (RUG) Program: Encouragement Grant via vote number Q.K130000.2638.10J04

6.0 REFERENCES

Burgess, M (1998). "Computer Immunology,"*Proceeding of the Systems Administration Conference*, (LISA-98), pp 283-297.

Cymtec Systems, Inc. (2012), *Scout Cloud-Enabled IDS Fact Sheet*.

Farhan Bashir Shaikh and Sajjad Haider (2011). "Security Threats in Cloud Computing," *6th International Conference on Internet Technology and Secured Transactions*, IEEE, pp 214-219.

Forrest, S, Perelson AS, Allen L and Cherukuri R (1994). "Self–Nonself Discrimination In A Computer," *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, IEEE pp 202.

Kephart, J (1994). "A Biologically Inspired Immune System For Computers," *Proceedings of the Fourth International Workshop on Synthesis and Simulatoin of Living Systems, Artificial Life* IV, pp 130-139.

Kim, J, Bentley P. J, Aickelin U, Greensmith, J, Tedesco, G and Twycross, J (2007). "Immune System Approaches to Intrusion Detection - a review," *Natural Computing*.

Mark Taylor, John Haggerty, David Gresty and David Lamb, (2011). "*Forensic investigation of cloud computing systems Network security*,"

Mell, P, and T. Grance (2011). "The NIST definition of cloud computing," *NIST special publication 800-145*.

Shanmugam, B. and N. B. Idris (2009). "Improved Intrusion Detection System Using Fuzzy Logic for Detecting Anamoly and Misuse Type of Attacks," *International Conference of Soft Computing and Pattern Recognition*, SOCPAR'09, IEEE.

S. Subashini, S., and V. Kavitha, 2011. A Survey on Security Issues in Service Delivery Models of Cloud Computing. *Journal of Network and Computer Applications*,Vol. 34(1): pp 1-11.

Somayaji, A, Hofmeyr, S and Forrest, S (1997). "Principles of a computer immune system,' *Proceeding of New Security Workshop*, Langdale, Cumbria, pp. 75-82.

Yassin, W., N.I. Udzir, Z. Muda, A. Abdullah and M.T. Abdullah (2012). "A Cloud-Based Intrusion Detection Service Framework," *International Conference on Cyber Security, Cyber Warfare and Digital Forensic* (CyberSec). IEEE.