

2nd International Conference on Communication, Computing & Security (ICCCS-2012)

Integrating Signature Apriori based Network Intrusion Detection System (NIDS) in Cloud Computing

Chirag N. Modi^{a,*}, Dhiren R. Patel^a, Avi Patel^b, Muttukrishnan Rajarajan^b

^aNIT Surat, INDIA

^bCity University London, UK

Abstract

One of the major security issues in Cloud computing is to detect malicious activities at the network layer. In this paper, we propose a framework integrating network intrusion detection system (NIDS) in the Cloud. Our NIDS module consists of Snort and signature apriori algorithm. It generates new rules from captured packets. These new rules are appended in the Snort configuration file to improve efficiency of Snort. It aims to detect known attacks and derivative of known attacks in Cloud by monitoring network traffic, while ensuring low false positive rate with reasonable computational cost. We also recommend the positioning of NIDS in Cloud. We present experimental setup and discuss the design goals expected from proposed framework.

© 2012 The Authors. Published by Elsevier Ltd. Selection and/or peer-review under responsibility of the Department of Computer Science & Engineering, National Institute of Technology Rourkela. Open access under [CC BY-NC-ND license](#).

Keywords: Cloud computing; Network based intrusion detection system; Snort; Signature apriori algorithm;

1. Introduction

Cloud computing is an innovative computing model providing resources and applications as a service over the Internet for satisfying the computing demand of the users. It provides Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Mell & Grance 2011). Exploitation of vulnerabilities existing in Cloud affects the confidentiality, availability and integrity of cloud resources and offered services. IDC survey concluded that security of Cloud services is the greatest challenge (Gens 2008). One of the major security issues in cloud computing is to protect against network attacks. The possible attacks at the network layer are IP spoofing, DNS poisoning, man-in-the-middle attack, port scanning etc. To prevent Cloud services from such intrusions, major Cloud providers (like Amazon, Windows Azure, Rack Space, Eucalyptus, Open Nebula etc.) integrate a firewall (*Cloud Computing Comparison Guide* N.d.). As shown in Fig. 1 (Lauzon N.d.), firewall protects the front access points of the system and is treated as the first line of defense. Therefore, insider attacks cannot be detected in Cloud. Few DoS or DDoS attacks are also too complex to detect using traditional firewalls. For instance, if there is an attack on port

* Corresponding author. Tel.: +91 9408883560.
E-mail address: cnmodi.956@gmail.com

80, the firewall cannot differentiate normal traffic from attack traffic (S. Beg & Mohsin 2010). Therefore, use of only traditional firewall to block all the intrusions is not an efficient solution.

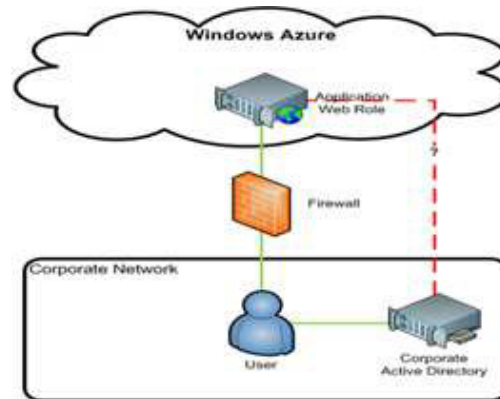


Fig. 1. Firewall in Windows Azure.

Another solution is to integrate a network based intrusion detection system (NIDS) in the Cloud. The efficiency of NIDS depends on parameters like detection technique (signature based or anomaly based), its positioning within the network (e.g. front end or back end) and its configuration (e.g. centralized or distributed) etc.

1.1. Our contribution

In this paper, we propose a framework that integrates NIDS in Cloud (offering IaaS). We also recommend different positioning of our NIDS module in the Cloud. Our main aim is to reduce impact of network attacks (known attacks as well as derivative of known attacks), while ensuring higher detection rate and lower false positive rate with an affordable computational cost.

The rest of this paper is organized as follows: Section 2 presents theoretical background and existing Cloud specific approaches, whereas the proposed framework is discussed in section 3. Section 4 discusses the design goals and shows the experimental setup with a conclusion and references at the end.

2. Theoretical Background and Related Work

2.1. Problem statement

The main objective is to design and integrate an efficient NIDS module that can detect intrusions in traditional as well as a virtual network in a cloud, while reducing false positive alerts with affordable computational cost.

2.2. NIDS in the cloud: existing approaches

Traditional NIDS uses signature based and anomaly detection techniques.

2.2.1. Signature based detection

It defines a set of rules (or signatures) used to decide that a given pattern is that of an intruder. Signature based systems are capable of attaining high levels of accuracy and minimal number of false positives in identifying intrusions. However, little variation in known attacks affects the analysis (D. J. Brown & Wang 2002). Signature based detection fails to detect unknown attacks or variants of known attacks. D. Stiawan *et al.* (D. Stiawan & Idris 2010) Presented issues regarding signature based system. In Cloud, the signature based detection technique can be used to detect external intrusions at the front end or to detect external/internal intrusions at the back end. Like traditional network, it fails to detect the unknown attacks.

2.2.2. Anomaly detection

It involves the collection of data relating to the behavior of legitimate use over a period, and then applies statistical tests to the observed behavior, which determines whether that behavior is legitimate or not. It has the advantage of detecting unknown attacks. T. Dutkevych *et al.* (T. Dutkevych & Tymoshyk 2007) provided anomaly based solution, which analyzes protocol based attack and multidimensional traffic. H. Zhengbing *et al.* (H. Zhengbing & Shirochin 2007) presented a lightweight intrusion detection system to detect the intrusion efficiently in real time. Anomaly detection technique can be used for Cloud to detect unknown attacks at different layers. However, large numbers of network level events makes difficult to monitor or control intrusions using anomaly detection technique in the Cloud.

2.3. Essential characteristics

NIDS should have the following characteristics for integrating it in the cloud.

2.3.1. Detection of network attacks on each layer

NIDS should be capable of detecting intrusions at each component like front end, back end or virtual machine (VM). It should be able to detect known attacks as well as unknown attacks.

2.3.2. Low computational cost and faster detection rate

In Cloud, high number of users are involved. So, high number of requests may turn into high traffic rate in Cloud. Therefore, NIDS should have faster detection at lower cost.

2.3.3. Low false positives

It can be defined as the number of false alerts generated by NIDS. This should be low for integrated NIDS in Cloud. It has been identified in (H. Zhengbing & Jumgi 2008) that in some specific situations, false positive alerts may be generated. E.g., alerts are generated in response to an ICMP flood, while in fact; there are several destination unreachable packets. Some unidentified packets generated by certain tools are alerted as attacks. Protocol violations cause false alerts.

2.3.4. Low false negatives

False negative can be defined as an inability of NIDS to detect the true intrusion. There are a number of reasons for causing the false negatives (H. Zhengbing & Jumgi 2008). If the traffic exceeds due to the ability of a switch, not all the network packets passing through such switch can be monitored. We need to keep very low false negatives in the Cloud. Otherwise, it results in the consumption of more resources in the Cloud.

2.4. Related work

As shown in Fig. 2 (S. Roschke & Meinel 2009), virtual machine (VM) compatible IDS architecture is composed of mainly two components: IDS management unit and IDS sensor. The IDS management unit consists of event gatherer, event database, analysis component and remote controller. Event gatherer collects malicious behavior identified by IDS sensor and stores in the event database. Event database stores information regarding captured events. Analysis component accesses the event database and analyze events as per the configuration. IDS-VMs are managed by the IDS Remote Controller which can communicate with IDS-VMs and IDS sensors. IDS sensors on VM, detects and reports malicious behavior and transmits triggered event to event gatherer. This approach is used to prevent the VMs from being compromised. However, this approach requires multiple instances of IDS.

A. Bakshi *et al.* in (Bakshi & Yogesh 2010) proposed an approach to detect DDoS attack in VM. In this approach, Snort is installed in the virtual switch to log network traffic into the database. To detect attack, logged packets are analyzed by Snort in real time. IDS determines the nature of the attack and notifies the virtual server. Then virtual server drops packets coming from the specified IP address. If attack type is DDoS, all the *zombie* machines are blocked. The virtual server then transfers targeted applications to other machines hosted by the separate data

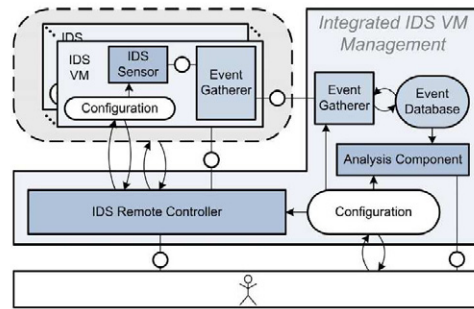


Fig. 2. The architecture of VM integrated IDS management.

center and updates routing tables immediately. Firewall placed on new server, blocks all the packets coming from an identified IP address.

C. Mazzariello *et al.* (C. Mazzariello & Canonoco 2010) presented an approach to detect intrusion in the eucalyptus Cloud. In this approach, Snort is deployed on the cloud controller (CC) as well as on physical machines (hosting VMs) to detect intrusion in the external networks. It is a fast and cost effective solution. However, it cannot detect unknown attacks.

In cooperative agent based approach (C. C. Lo & Ku 2008), individual NIDS module is deployed in each Cloud region. In case of intrusion detection, it drops attacker's packets, then sends alert message to other region. Alert clustering module collects alerts produced by other regions. Whether this alert is true or false, is determined after calculating the severity of collected alerts. This approach is suitable for preventing Cloud from a single point of failure caused by DDoS attack. However, the computational effort is very high.

A.V. Dastjerdi *et al.* (Dastjerdi, Bakar & Tabatabaei 2009) proposed scalable, flexible and cost effective method to detect intrusion for Cloud applications regardless of their locations using mobile agent (MA). As shown in Fig. 3, VMs are attached to MA which collects evidences of an attack from all the attacked VMs for further analysis and auditing.

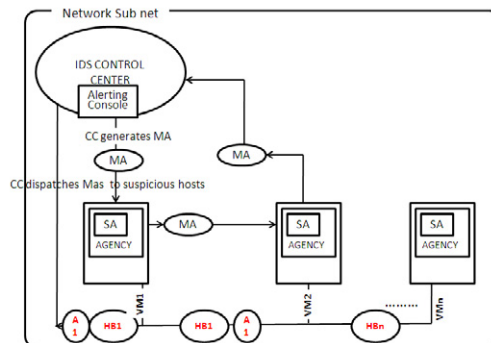


Fig. 3. Intrusion detection using mobile agent.

3. Proposed Framework: Signature Apriori based NIDS in Cloud

3.1. Design goals

- Detection of derivative of known attacks
- Low computational cost and faster detection rate
- Low false positives

Some attack signatures are formed from known attacks and can be found in the payload of captured packets. For instance, Code Red I and Code Red II are known "worms", which spread themselves through a network. Signatures in payload for both the attacks are:

Content: "|2F6465661756C742E6964613F4E4E4E|" (Code Red I).

Content: "|2F6465661756C742E6964613F585858|" (Code Red II).

In both signatures, the common string is "2F646566 1756C74 2E696461", which shows that Code Red I and Code Red II are derived form of "2F646566 1756C74 2E696461". Proposed NIDS produces and updates such signatures in Snort.

3.2. Building a solution framework

As shown in Fig. 4, Cloud can be viewed with two ends viz; Front end and Back end (Processing Server). Cloud users are able to communicate with Cloud via front end. Front end is connected to both external network as well as internal network. Processing server consists of computer hardware and software that are designed for the delivery of services. It processes the user's query and executes it for allowing to access VM instances. Internal network (virtual network) is designed for VM instance interconnectivity. E.g., in Amazon's eucalyptus Cloud, each VM instance has two network IPs named public IP and private IP (D. Nurmi & Zagorodnov 2008). VMs can communicate directly using private network.

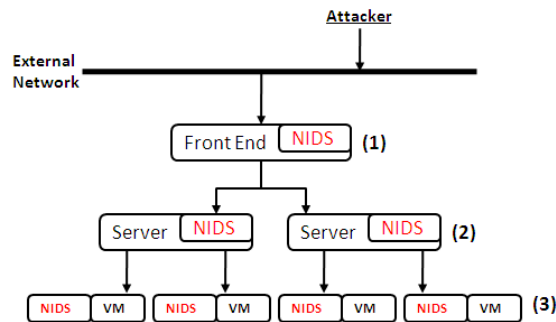


Fig. 4. Different positioning of NIDS in Cloud.

There may be different positioning of NIDS in Cloud as shown in Fig. 4: Positioning NIDS module on the front end of Cloud helps to detect network intrusions at the external network of Cloud. However, it is not able to detect internal intrusions. Positioning NIDS module on the processing server helps to detect internal as well as external intrusions. Large number of packets passing through server affects the efficiency of integrated NIDS. Integration of NIDS on each VM helps user for detecting intrusion on his/her VM. Such configuration requires multiple instances of NIDS, which makes complex management since VMs are dynamically migrated, provisioned or de-provisioned.

3.3. Design of NIDS module and its working

As shown in Fig. 5, we combine Snort (*Snort-Home page* N.d.) and signature apriori algorithm (H. Zhengbing & Jungi 2008) in our NIDS module. The network may be external network or internal network. We use Snort for detecting network intrusions, whereas the signature apriori algorithm is used to generate new possible signatures from partially known signatures. The Snort and signature apriori algorithm are chosen due to their following characteristics:

3.3.1. Snort

It uses a signature based detection technique. Snort is configurable, free, widely used, can run on multiple platforms (i.e. GNU/Linux, Windows) and is constantly updated. It captures network data packets and checks their content with the predefined patterns for any correlation. The detection engine of Snort allows registering, alerting and responding to any known attack. In inline mode of Snort, the functionality of Snort is extended for active defense capability.

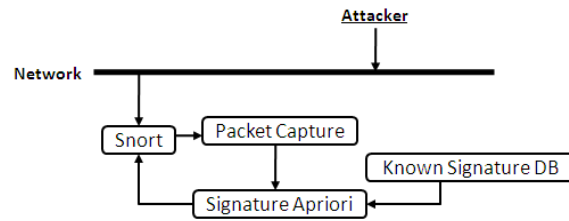


Fig. 5. Design of NIDS module.

3.3.2. Signature Apriori

It takes to capture packets and partially known signatures as input. As an output, it generates new attack signatures. These new signatures are used in Snort for detecting the derivatives of known attacks. In the proposed design, signature apriori is combined with Snort for accurate and efficient detection. The proposed method has less training time compared to other classification techniques given in (Han & Kamber 2006).

Workflow of our NIDS module is shown in Fig. 6. Network packets are captured from network (external or internal) using Snort. Snort will monitor those network packets and allow/deny them based on the configured rules. Also, captured packets, partially known attack signatures (stored in known signature database) and support threshold are given as input to the signature apriori algorithm. Security administrator updates known signature database. Using given input, signature apriori generates new possible signatures and updates them as rules in Snort. So, derivative attacks can be detected by Snort. In such a way, our design can be used to detect known attacks as well as derivative attacks. We discuss an example demonstrating signature generation in our design as follows:

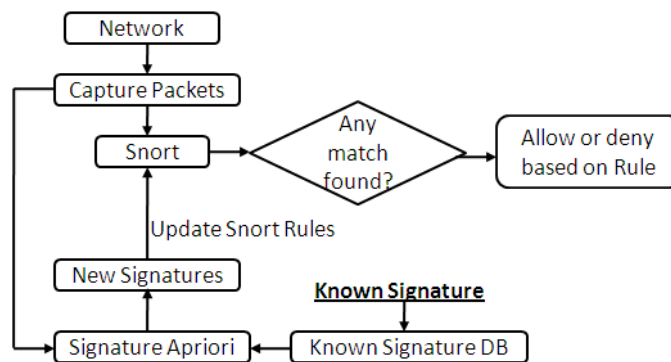


Fig. 6. Working of NIDS module.

In Table 1, a dataset of captured packets (with packet ID and its contents) is given. Suppose 0.7 as a minimum support and "C D E" as a partially known attack signature specified by the security administrator.

Table 1. Sample dataset of network packet content.

Packet ID	Payload contents
1	M A B C E F G P Q
2	M N A B C D E F G
3	A B C D E F G Q
4	J A B C D E F G
5	N A B C D E F G Q
6	P Q I

Signature apriori first finds frequent-1 itemset by calculating support count for each item. Here, candidate itemset is as follows: A, B, C, D, E, F, G, Q, M, N, P, J, I. After the first iteration, signature apriori finds frequent-1 itemset

satisfying minimum threshold: A: $5/6=0.83$, B: 0.83, C: 0.83, D: 0.83, E: 0.83, F: 0.83, G: 0.83. Now, algorithm joins frequent-1 items to the right side of known signature and generates candidate set: C D E A, C D E B, C D E C, C D E D, C D E E, C D E F, C D E G. From this candidate set, it finds frequent itemsets: C D E F: 0.83. Again it joins frequent-1 itemset to a generate candidate set and finds frequent itemsets and continues until minimum support satisfies. Finally, it generates C D E F G. Now, algorithm joins frequent-1 itemset to the left side of last produced signature and generates candidate itemset to find frequent itemset. Finally, it produces A B C D E F G. Generated frequent itemsets at each iteration are taken as signatures of attacks. So, possible attack signatures are: C D E F, C D E F G, A B C D E F G. However, as shown in (H. Zhengbing & Jumgi 2008), longer signatures give more accuracy than the shorter string for detection. So, "A B C D E F G" can be used as a new derivative signature, which gives better accuracy.

4. Experimental Setup and Analysis

4.1. Experimental setup

We have used Eucalyptus (D. Nurmi & Zagorodnov 2008), (an open source Cloud) installed on the Ubuntu Linux operating system. As shown in Fig. 7, Cloud controller (CLC) and cluster controller (CC) are installed on the front end; whereas node controller (NC or processing server) is installed on back end. NIDS (Snort and signature apriori (SA) algorithm) is installed on each NC. For testing purpose, we allow all types of traffic by opening all the ports in Eucalyptus. Scapy (Scapy N.d.) is used to generate custom packets having signature of attacks and sending them to the front end. Wireshark (WS) (Wireshark N.d.) installed on the front end and back end of Cloud is used to monitor traffic passing through them. We have used MySql database (DB) to log detected malicious packets.

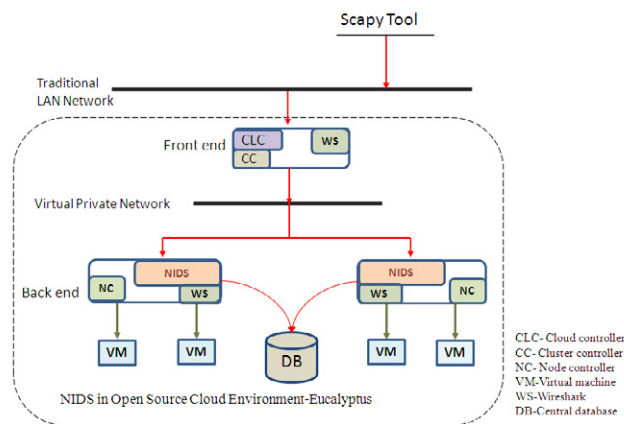


Fig. 7. Experimental setup of proposed framework.

4.2. Theoretical analysis

4.2.1. Detection of known attacks as well as derivative of known attacks at each layer of the Cloud

In the proposed framework, known attacks are detected by Snort, whereas the signatures of derivative attacks are formed using signature apriori and updated as rules in Snort, which helps to detect derivative of known attacks. Derivative DoS attacks can also be detected, since the frequency of the signature is considered at generation time. It secures Cloud components (like front end, processing servers, VMs etc.) from network intrusions.

4.2.2. Low false positive alert rate

Snort has low false positive alert rate of known attacks since signature based technique is used, where variation of known attacks are detected by rules configured based on new generated signature. Here, signature having longer string

is considered as attack signatures since the probability of the shorter string in normal traffic is high (H. Zhengbing & Jumgi 2008). Thus, proposed framework has lower false alarm rate.

4.2.3. Low computational cost

It has lower computational cost since once rules are generated by using signature apriori, they are not generated again. It uses signature based technique which has low computational cost. Moreover, proposed framework does not require a high number of NIDS instances, as in (S. Roschke & Meinel 2009).

5. Conclusion

Existence of vulnerabilities in Cloud computing allow intruders to affect the confidentiality, availability and integrity of cloud resources as well as services. Detection of DoS/DDoS attack and other network level malicious activities are major security concerns in the Cloud. To address this issue, integration of only firewall in the cloud is not an efficient solution. Our proposed solution framework (integrating NIDS into Cloud) can be used to detect network attacks (known attacks as well as derivative of known attacks) at the front end as well as the back end of Cloud environment (i.e IaaS). It aimed to achieve low false positive alarm rate within reasonable computational cost.

References

- Bakshi, A. & B. Yogesh. 2010. Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine. In *Second International Conference on Communication Software and Networks*. pp. 260–264.
- C. C. Lo, C. C. Huang & J. Ku. 2008. Cooperative Intrusion Detection System Framework for Cloud Computing Networks. In *First IEEE International Conference on Ubi-Media Computing*. pp. 280–284.
- C. Mazzariello, R. Bifulco & R. Canonoco. 2010. Integrating a network IDS into an Open source Cloud computing. In *Sixth International conference on Information Assurance and Security (IAS)*. pp. 265–270.
- Cloud Computing Comparison Guide. N.d. <http://www.webhostingunleashed.com/whitepaper/cloud-computing-comparison/>.
- D. J. Brown, B. Suckow & T. Wang. 2002. A Survey of Intrusion Detection Systems. Technical report Department of Computer Science, University of California, San Diego.
- D. Nurmi, R. Wolski, C. Grzegorzczak G. Obertelli S. Soman L. Youseff & D. Zagorodnov. 2008. Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems. Technical report UCSB Computer Science Technical Report Number 2008-10: .
- D. Stiawan, A. H. Abdullah & M. Y. Idris. 2010. The Trends of Intrusion Prevention System Network. In *2nd International Conference on Education Technology and Computer (ICETC)*. Vol. 4 pp. 217–221.
- Dastjerdi, Amir Vahid, Kamalrulnizam Abu Bakar & Sayed Gholam Hassan Tabatabaei. 2009. Distributed Intrusion Detection in Clouds Using Mobile Agents. In *Proceedings of the 2009 Third International Conference on Advanced Engineering Computing and Applications in Sciences*. ADVCOMP '09 pp. 175–180.
- Gens, F. 2008. "IT Cloud Services User Survey, pt.2: Top Benefits & Challenges." <http://blogs.idc.com/ie/?p=210>.
- H. Zhengbing, L. Zhitang & W. Jumgi. 2008. A Novel Intrusion Detection System (NIDS) Based on Signature Search of Data Mining. In *WKDD First International Workshop on Knowledge discovery and Data Mining*. pp. 10–16.
- H. Zhengbing, S. Jun & V. P. Shirochin. 2007. An Intelligent Lightweight Intrusion Detection System with Forensic Technique. In *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. pp. 647–651.
- Han, J. & M. Kamber. 2006. *Data Mining Concepts and Techniques*. Second ed. Morgan Kaufmann Publishers.
- Lauzon, Vincent-Philippe. N.d. "Departmental Application Migration to Azure - Part 2 - ADFS Installation." <http://vincentlauzon.wordpress.com/2010/06/02/departamental-application-migration-to-azure-part-2-adfs-installation/>.
- Mell, P. & T. Grance. 2011. "The nist definition of cloud computing (draft)." <http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145.cloud-definition.pdf>.
- S. Beg, U. Narul, M. Ashraf & S. Mohsin. 2010. "Feasibility of Intrusion Detection System with High Performance Computing: A Survey." *International Journal for Advances in Computer Science* 1(1).
- S. Roschke, C. Feng & C. Meinel. 2009. An Extensible and Virtualization Compatible IDS Management Architecture. In *Fifth International Conference on Information Assurance and Security*. pp. 130–134.
- Scapy. N.d. <http://www.secdev.org/projects/scapy/>.
- Snort-Home page. N.d. <https://www.Snort.org/>.
- T. Dutkevych, A. Piskozub & N. Tymoshyk. 2007. Real-Time Intrusion Prevention and Anomaly Analyse System for Corporate Networks. In *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. pp. 599–602.
- Wireshark. N.d. <http://www.wireshark.org/>.