Detection of Distributed Attacks in Hybrid & Public Cloud Networks

Syed Raheel Hassan #*1, Julien Bourgeois #2, Vaidy Sunderam *3, Li Xiong *4

#FEMTO-ST Institute, UMR CNRS 6174, University of Franche-Comte (UFC)
1 Cours Leprince-Ringuet, 25201 Montbeliard, France
1raheel.hasan@univ-fcomte.fr
2julien.bourgeois@femto-st.fr

*Department of Mathematics & Computer Science, Emory University
400 Dowman Dr., W401 Atlanta, GA, USA
3vss@emory.edu 4lxiong@mathcs.emory.edu

Abstract— **In this paper early detection of distributed attacks are discussed that are launched from multiple sites of the hybrid & public cloud networks. A prototype of Cloud Distributed Intrusion Detection System (CDIDS) is discussed with some basic experiments. The summation of security alerts has been applied which helps to detect distributed attacks while keeping the false positive at the minimum. Using the summation of security alerts mechanism the attacks that have slow iteration rate are detected at an early stage. The objective of our work is to propose a Security Management System (SMS) that can detect malicious activities as early as possible and camouflaging of attacks under the conditions when other security management systems become unstable due to intense events of attacks.**

I. Introduction

Cloud computing provides Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). There exist multiple types of clouds such as community, private, public and hybrid clouds. Community clouds are formed by multiple organizations which are working for the same objective. Private cloud is owned by one administrative domain, this type of cloud network is used by organizations who wants to have full control of the cloud resources. Private cloud is not used by many users as the benefits are very limited. Public cloud are the most common and easily available place for user to use resources either free or pay as per usage. Hybrid clouds are the combination of community or public clouds. The member of these clouds can get benefits of both the cloud networks. It provides the users a large range of resources available to use. There exists many cloud service provider and they are growing as the technology is becoming mature. The most prominent are the Amazon Elastic Compute Cloud (Amazon EC2) [1] or S3 [2], Google cloud services [3], Eucalyptus [4], IBM smart cloud [5] and Opennebula [6].

One of the security concern in the cloud computing, is the handling of the data that is going to be placed at the service provider's network. The data placed on the cloud can be misused or compromised and the owner does not even know about the incident. A security management systems called Cloud Distributed Intrusion Detection (CDIDS) has been proposed in this paper. CDIDS's separate policies can be applied on the data in the form of security rules. The security rules generate alarms if the security policy is violated by the cloud service provider, by the owner itself or by any of its staff members. The proposed solution can also helps in resolving issues of privacy in the cloud infrastructure. In this paper more details of the security issues in cloud networks are discussed with their proposed solution.

The remaining part of the paper consists of VI sections. Section II is the related work, section III highlights the existing problems, section IV presents the proposed solution, section V consists of the experiments, and section VI concludes and outlines the future development.

II. Related Work

The most recent issues are discussed by Balduzzi et al. [7] where they highlighted the security issues present in the public virtual images. They performed vulnerability tests on 5000 virtual machine images available in four different data centers of Amazon [1] and reported several security issues, some of them are:

(i) The confidential files were deleted while preparing the virtual machine image but these file are easily recoverable such as password files, SSH private keys, PGP private keys, etc.
(ii) Discovered instances of SSH, different services and Web.
(iii) History of files of VNC, MySql, DNS, WebApp, and SQL.

Bugiel et al. [8] also highlighted the similar issues discussed by Balduzzi et al. [7], but they performed experiments on 1255 Amazon images and the scope of their experiments was limited in covering security issues. The main focus of both the findings is to emphasis that there exist some serious security threats in cloud computing infrastructures. Garfinkel and Rosenblum [9] highlighted the use of third party virtual images and their security issues. They also discussed other security issues exist in user generated virtual images. Glott et al. [10] highlighted the security issues that occurs when the virtual images are shared within multiple users in cloud infrastructure. They proposed some assessments to find out the vulnerabilities present in the virtual image. Ristenpart et al. [11] only presented the introduction of side channel attacks in cloud computing networks. Bleikertz at al. [12] used graph theory techniques to deploy virtual machine images in AmazonEC2 infrastructure. The objective of their work is to focus on the security issues that are present at the infrastructure level which is a different approach as others focus more on the virtual images security. Their propositions are based on configuring network and setting the security policies properly.

III. Design Goals

In this section four types of problems are discussed namely; early detection of attacks, scalability and fault-tolerance, an intensive attack which last for very short period of timed, and minimization of false positives.

(I) **Early detection of Attacks:** In large computing infrastructures such as grids and clouds early detection of attacks is very important. Due to the nature of these networks where the network is shared by many users dispersed globally. Those attacks that are detected with a slight delay can cause very serious consequences for all the users. As we know that users from different organization can have access to cloud computing infrastructures. It is highly possible that some of the users that use the cloud and its services are not very secure within there network. It includes numerous factors such as vulnerabilities present in their machines, lack of anti-virus and firewalls in the network, no or poor security policies are in place in their local network. These users are the main source which can be exploited by attackers and can be used as zombies. They can become the cause of attack distribution to other secure users, which are part of the organization that has spend a lot of money and resources to make their network secure. One possible vulnerable area which is discussed in the related work are the reused of the virtual machines images that are shared among the users. Although the users for highly important experiments does not uses the virtual images of other users but there are chances that the virtual images that are used by other users does have some back-doors open in for form of root-kits. These back-door are used by the attackers to collect sensitive information of the other participating members of the network.

(II) **Scalability and Fault-Tolerance:** In large size computing infrastructure scalability is a key factor because the size of the networks always grows dynamically. This increase in the size leads to condition where the Security Management System (SMS) that manages the security of that network becomes overloaded by huge number of security alerts. Current SMSs struggle a lot under intensive attacks discussed by Raheel et al. in [13] and [14]. Their experiments show that the performance of the SMSs degrade after certain period of time. As large networks are distributed to many locations therefore the SMS is also distributed in order to monitor every site of the network. Fault-Tolerance is mandatory when any component of the SMS fails to works or targeted by the attackers. If the replacement of that component is not provided on time the SMS cannot not detect the attacks that are in place at the premises of that local site of the cloud network.

(III) **An intensive attack which last for very short period of time:** When attackers are distributed across different sites of the network they use slow timed pace attacks. The reason to use low iteration of the attempts is to hide the attacks from the SMS. For example a brute force attack that has a dictionary which contains thousands of passwords. If the attacker uses it from one place by one instance, he needs to keep trying at a high iteration rate to break the password of the machine. There are high chances that due to the high number of attempts the attacker is identified by the SMS. If the attack uses slow iteration rate than it might take too long to get success. Therefore the best approach is to divided the dictionary of passwords in small chunks of multiple files. The attacker uses multiple instances of the attacks and uses these small dictionary of passwords simultaneously. The attacker slows down the number of attempts from each instance and get the same iteration rate if combined together. This technique is very common and quite effective if the SMS has some threshold defined to generate the attack alarm. The attacker keep the iteration of the attempts below this threshold value and gets success in its attacks.

(IV) **Minimization of false positives:** The events are collected and analyzed in order to produce an alert. For generating an attack alarm there are two possibilities. First; the SMSs mostly use some threshold value before generating an attack alarm. This threshold triggers an alarm if any specific alert reaches a defined limit of iterations. Second; the SMS has some defined security rules, On the basis of these rules, SMS decides whether it is an attack or a normal activity. The wrong attack alarm can be reduce if the common attack alerts are combined at a specific time. This mechanism is effective in minimizing the false positive rates and enhances the performance of the SMSs.

IV. Proposed Solution

To address all the above mentioned problems a SMS called Cloud Distributed Intrusion Detection System (CDIDS) has been proposed. Only the overview of the CDIDS has been discussed in this paper. Figure 1 is the internal view of CDIDS. It shows that CDIDS is based on log collection model. CDIDS collects logs from multiple collectors that are distributed and placed across the cloud infrastructure. Collectors are the modules that are programmed to receive logs from computing elements present in the network. Similarly computing elements are configured to send their logs to the collectors. Several collectors are distributed across all the local sites of the network. The number of collectors can be increased or decreased depending upon the numbers of logs generated in that local site. The collectors analyze the logs to detect the attack incidents. If some incidents are found, they are marked as alerts and forwarded to the log manager. The main controller and the analyzer of the systems is the log manager that holds the security alerts which come from different cloud networks. The log manager formates the received local alerts and correlate them to report whether it is really an attack incident or not. It further analyzes all the local alerts received to detect the distributed attacks lunched on multiple local sites of the network. The detected attacks are formatted in a standard formate that is defined in the CDIDS. All the detected alerts with a specified formate of CDIDS are forwarded to match with the general security rules that are written by the security manager. In general security rules, if the security alerts are matched with any of the defined rule. One alarm generates containing all the necessary information about the attack, its re-occurrence, the time it was started and the elapsed time. At the final stage a further analysis of the attack is done to calculate its intensity. Once all the parameters are marked it is saved into the cloud service provider's database for further forensic analysis.

Figure 2 represents the overview of the cloud security management system which composes of multiple public & hybrid clouds. Each cloud consists of several Administrative Domains (AD) which are further divided into many local sites. It shows that how events are collected and correlated to detect the attacks which can occur in different cloud networks. The CDIDS helps to handle all the above mentioned problems. It detects the attacks at a very early stage by summation of security alerts globally. Summation of alerts can be done securely by using Secure Multi-party Computation (SMC). The SMC is first proposed by Andrew Chi in [15]. In an SMC setting [16][17], a number of participants, each having a piece of private data, wishes to compute the value of a public function. A protocol is secure if the computation yields, to all participants, no more information than their own inputs and the final result. In the proposed approach, secure sum and secure comparison protocols can be used to sum the number of security alerts from individual administrative domains and to determine if it exceeds a given threshold. The secure sum and secure comparison protocols can be implemented using the SEPIA framework [18] based on Shamir's secret sharing scheme. It even detects those attacks that lasts for very short period of time. CDIDS has a scalable architecture and it is fault-tolerant. In order to minimize the false positives and accurately detect the attacks a distributed security alert summation technique has been adopted. It helps in detecting the attack by adding all the reported security alerts occurred during the same period of time within all the member of the cloud computing network. In CDIDS security alerts sharing mechanism between intra cloud can be further improved using privacy preserving data release techniques such as those based on k-anonymity [22][23] and differential privacy [19][20] [21].

V. Experiments for Optimizing Detection of Distributed Attacks in Cloud Computing Networks

In this section three types of experiments are performed in order to handle the problems discussed above. CDIDS which is proposed in this paper can adopt the techniques shown in this section. Figures 3, 4 and 5 are the results achieved by simulating the characteristics of the SMS in our testbed.

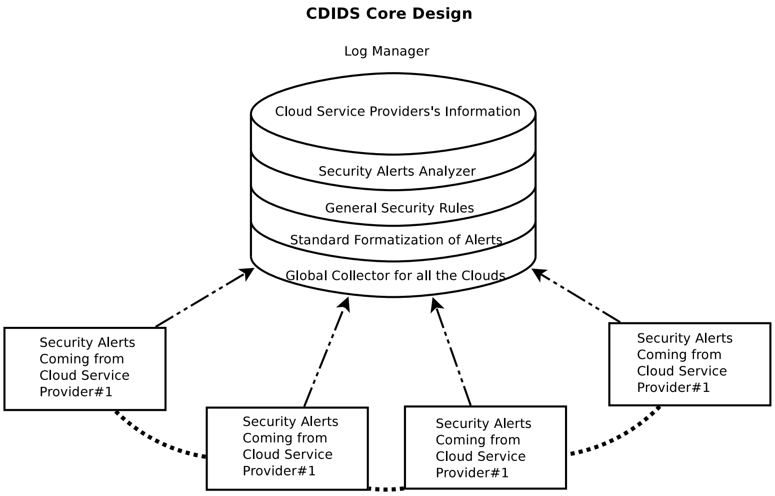
CDIDS Core Design
Log Manager
Cloud Service Providers's Information
Security Alerts Analyzer
General Security Rules
Standard Formatization of Alerts
Global Collector for all the Clouds
Security Alerts Coming from Cloud Service Provider#1
Security Alerts Coming from Cloud Service Provider#1
Security Alerts Coming from Cloud Service Provider#1
Security Alerts Coming from Cloud Service Provider#1

Fig. 1. Cloud Distributed Intrusion Detection (CDIDS) Core Design

A. *Smurf Attack detection*

Smurf attack is a type of DoS attack in which the attacker uses spoofed IP addresses in order to generate huge traffic at the victim machine. In smurf attack one attacker or group of attackers use large amount of ICMP packets to a specific or broadcast IP addresses. In this attack scenario the group of attackers launched the attack simultaneously on four Cloud Network (CN). They spoofed their IP addresses which makes it harder to detect early by the security management system. Using the summation of the sites the attacks are detected earlier. Multiple attackers use multiple commands of hping to launch the attacks, which are as under,

Attacker1: **hping3 -a spoofed_IP_address -i u1 -S victim_IP_address**
Attacker2: **hping3 -a spoofed_IP_address -i u10 -S victim_IP_address**
Attacker3: **hping3 -a spoofed_IP_address -i u100 -S victim_IP_address**
Attacker4: **hping3 -a spoofed_IP_address -i u1000 -S victim_IP_address**

The parameter of the hping command are, ”-a” is used to set any IP address, ”-i u” is the delay in transmitting another packet in micro seconds and ”-S” is used to set the SYN TCP flag.

Figure 3 shows the early detection of Smurf attack using the security alerts summation mechanism. The received alerts from the members of the cloud are further analyzed at each CN to detect distributed attacks. Three thresholds are set globally to detect the distributed attacks. The reason to set these thresholds is to detect the attacks as early as possible

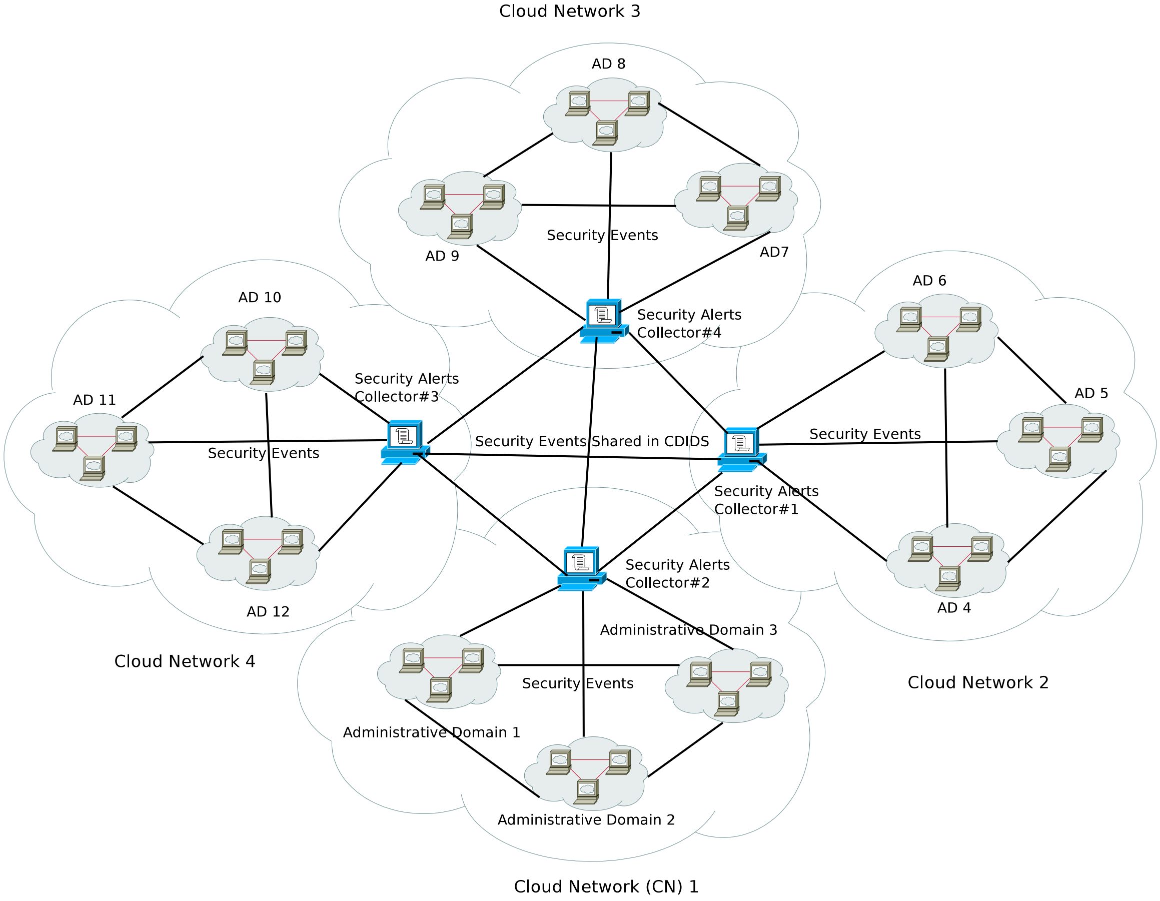

Fig. 2. General View of Intra-public Cloud Architecture

and to minimizing the false positives. The threshold values are adjustable according to the size of the network and the capacity to handle and store the security alerts globally. The low threshold is set to 100000 number of security alerts. The medium threshold is set to 250000 number of alerts. The high threshold is set to 400000 number of alerts. Using the low threshold the Smurf attack is detected on CN 1,3&4 between 1 to 2 minutes. The medium threshold detects the Sumurf attack on CN 3&4 is detectable between 3 to 4 minutes whereas on CN 2,3&4 the same attack is detected in between 1 to 2 minutes. The high threshold can only detect the summation of CN 2,3&4 at the fourth minute. This experiment shows that early detection of attacks explained in problem I can be improved by using the summation of security alerts. It can also help the SMS to minimize the false positives discussed in problem IV, because the alarm of attack that has the iteration rate in thousands of alerts proves that SMS has detected a real attack.

B. SYN Flooding Attack Detection

TCP protocol uses three-way handshake mechanism which is vulnerable to SYN flood attacks. (i) Attacker sends ”SYN” to victim. (ii) Victim sends ”SYN-ACK” back to the attacker. (iii) Attacker does not sends ”ACK” and keeps sending ”SYN” packet to the victim. The SYN flood attack is one kind of DoS attack where high number of ”SYN” packets are send by the attackers to the victim machine. The victim machine allocates resources for each request sent by the attackers. The victim machine sends back the ”SYN-ACK” to the source IP of the attackers. The attackers use the spoofed IP addresses which does not exists in the network. This results in no ”ACK” response from the attackers. The attackers continue sending high amount of ”SYN” packets to the victim machine and victim waits for the ”ACK” request. After some time these SYN packets sent by the attackers consume all the resources of the victim machine and makes it unstable which results in blocking all the requests coming from legitimate users. The SYN attack can halt the entire network operations if uses the broadcast address mixing with IP spoofing.

Figure 4 shows the SYN attack case, where the thresholds are set to detect the instability in the security management system. Four CNs are shown in the graph which are the member of one cloud network. The curves show that after passing of 2 minutes the CNs does not detect the attacks at the same rate and continue to work at a constant rate. This case occurs when the CN does not have much resources to handle the intense attacks. Here the summation of alert

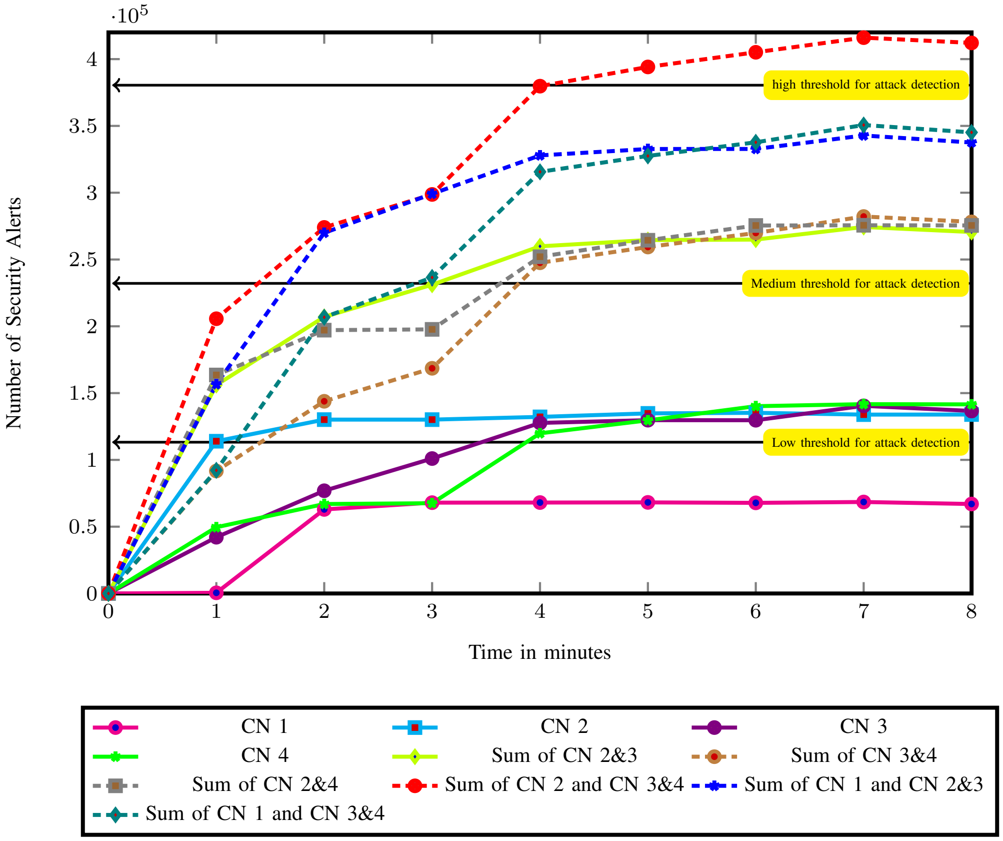

Fig. 3. Detection of Smurf Attack

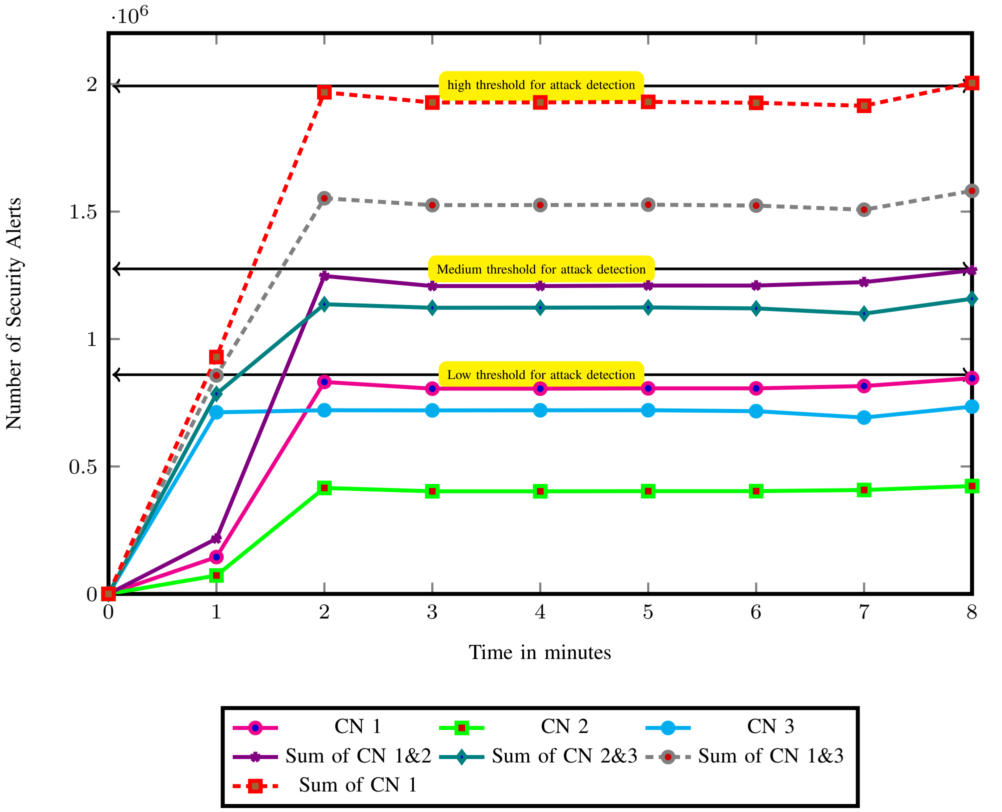

Fig. 4. Detection of SYN Attack

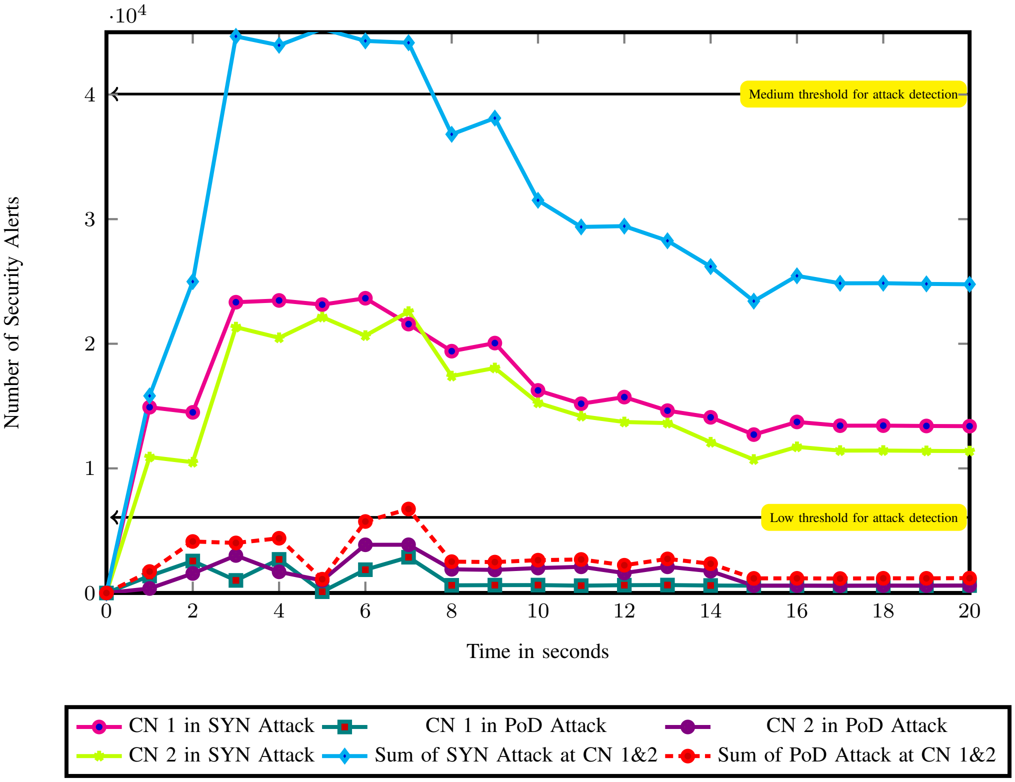

Fig. 5. Multiple Attack Detection in seconds

mechanism is very useful because it detects the attacks even, when the SMSs are struggling. At the low threshold level the sum of CN 1&2 and the sum of CN 2&3 are detected in between 1&2 minutes. The medium alert threshold level detects the summation of alerts of the CN 1&3 and sum of CN 1,2&3 in between 1&2 minutes. The high alert threshold level only detects the summation of alerts of the CN 1,2&3 after 8 minutes. The hping command which is used to launch the attack by the attackers is given below,

hping3 –syn –destport 80 -i u1000 Victim_Machine

The parameter of the commands are, ”–syn” is to set the SYN tcp flag, ”–desport 80” is to set the port number of the victim and ”-i u1000” is to send the packet after the delay of 1000 micro seconds. IPtable rules to detect the malicious packets is,

iptables -A INPUT -d 0/0 -s0/0 -p tcp –tcp-flags SYN,ACK,FIN,RST SYN -j LOG –log-prefix ”SYN_ATTACK ”

This rule logs the packets that are coming from any source to any destination using TCP protocol having any of the bit set SYN,ACK,FIN,RST. It saves every attempt of this type with the tag ”SYN ATTACK” in the system logs. This experiment handles the instability issues discussed in problem II and shows that the attacks are detected before the SMS becomes unstable.

C. Distributed SYN and PoD Attack Detection in Seconds

In this attack multiple attackers use SYN and PoD attacks together. Figure 5 shows the behavior of the SYN and PoD attack. The commands that are used for these attacks are,

Attacker1: **hping3 –syn –destport 80 -i u10 Victim_Machine**
Attacker2: **hping3 –syn –destport 80 -i u100 Victim_Machine**
Attacker3: **ping -s 64000 -i 0 IP**
Attacker4: **ping -s 65000 -i 0 IP**

The parameter of the Ping command are, ”-s” defines the packet size, ”-i” is the interval between two packets. Here ”0” means it is set to flood mode.

In this scenario the attackers increase and decrease their attack intensity in order to camouflage their malicious activities from the SMS. Therefore the attack detection is more optimized by setting the thresholds in seconds. Multiple attackers launches the SYN and PoD attack at the CN 1&2. The objective of these attacks is to destabilize the security management system and hide the real attacks such as Brute Force. The graph shows that low threshold detects the distributed PoD attack on CN 1&2 in between 6 to 7 seconds. The medium threshold detects the distributed SYN attack on CN 1&2 in between 3 to 7 seconds. This experiment addresses problem III and shows that camouflaging the malicious activities are now detectable.

VI. Conclusion & Future Work

In this paper the internal architecture of CDIDS and its core features are presented. The general view of CDIDS has been explained that shows it is scalable and fault-tolerant. Some experiments are conducted to show that how the early detection of distributed attacks is possible. Few advantages are highlighted such as minimization of false positives and detection of attacks even when the SMS become unstable. The summation of security alerts method helps in detection of very powerful attacks which last for very short period of time.

Future modification in the design of the CDIDS needs to be done, some of the propositions are:

(i) New boxes should be introduced at each service, that means one box for infrastructure, one for platform and one for software.

(ii) These boxes must be programmed to handle the specific security issues occurred at each service level.

(iii) For every service there must be a separate analyzer which communicates with the boxes located in its vicinity.

(iv) If needed the analyzers can also collaborate with each other to detect the attacks that are launched using all the three service levels.

(v) The analyzers at each service level must report to the Main analyzer which will handle the security of the entire cloud.

(vi) There may exist multiple dedicated boxes which are allowed to share the security information with each other.

(vii) The manager of all the boxes will be the Cloud Box that will present the global view of the security.

Acknowledgment

Thanks to the Laboratory of Computer Science University of Franche-Comte France, the Higher Education Commission and Quaid-e-Awam University of Engineering, Science & Technology Pakistan for supporting our work financially. The Department of Computer Science of the University of Emory USA for technical assistance and collaboration.

References

[1] "Amazon elastic compute cloud (amazon ec2)," Available at : http://aws.amazon.com/ec2/, 2012.

[2] "Amazon simple storage service (amazon s3)," Available at : http://aws.amazon.com/s3/, 2012.

[3] "Google?s cloud services," Available at : http://www.google.com/enterprise/cloud/, 2012.

[4] "Eucalyptus is the world's most widely deployed cloud computing software platform for on-premise (private) infrastructure as a service clouds," Available at : http://www.eucalyptus.com/, 2012.

[5] "Smartcloud is the ibm vision for cloud computing," Available at : http://www.ibm.com/cloud-computing/us/en/, 2012.

[6] "Opennebula is an open-source project," Available at : http://opennebula.org/cloud:usingit, 2012.

[7] M. Balduzzi, J. Zaddach, D. Balzarotti, E. Kirda, and S. Loureiro, "A security analysis of amazon's elastic compute cloud service," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 1427–1434. [Online]. Available: http://doi.acm.org/10.1145/2245276.2232005

[8] S. Bugiel, S. Nürnberger, T. Pöppelmann, A.-R. Sadeghi, and T. Schneider, "Amazonia: when elasticity snaps back," in *Proceedings of the 18th ACM conference on Computer and communications security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 389–400. [Online]. Available: http://doi.acm.org/10.1145/2046707.2046753

[9] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: security challenges in virtual machine based computing environments," in *Proceedings of the 10th conference on Hot Topics in Operating Systems - Volume 10*, ser. HOTOS'05. Berkeley, CA, USA: USENIX Association, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251123.1251143

[10] R. Glott, E. Husmann, A.-R. Sadeghi, and M. Schunter, "Trustworthy clouds underpinning the future internet," in *IEEE Signal Process. Lett.'11*, 2011, pp. 209–222.

[11] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 199–212. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653687

[12] S. Bleikertz, M. Schunter, C. W. Probst, D. Pendarakis, and K. Eriksson, "Security audits of multi-tier virtual infrastructures in public infrastructure clouds," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, ser. CCSW '10. New York, NY, USA: ACM, 2010, pp. 93–102. [Online]. Available: http://doi.acm.org/10.1145/1866835.1866853

[13] S. R. Hassan, J. Pazardzievska, and J. Bourgeois, "Minimization of security alerts under denial of service attacks in grid computing networks," in *The Int'l Conf. Grid Computing and Applications (GCA11)*, WORLDCOMP11. Las Vegas, USA: CSREA Press, July 2011, pp. 44–50.

[14] S.R.Hassan, J. Pazardzievska, and J. Bourgeois, "Fast attack detection using correlation and summarizing of security alerts in grid computing networks," *The Journal of Supercomputing*, p. 24, 2012.

[15] A. C.-C. Yao, "Protocols for secure computations (extended abstract)," in *FOCS*, 1982, pp. 160–164.

[16] O. Goldreich, *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[17] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *The Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59–98, 2009.

[18] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos, "Sepia: Privacy-preserving aggregation of multi-domain network events and statistics," in *USENIX Security Symposium*, 2010.

[19] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys*, vol. 42, no. 4, pp. 14:1–14:53, June 2010.

[20] C. Dwork, "Differential privacy: a survey of results," in *Proceedings of the 5th international conference on Theory and applications of models of computation*, ser. TAMC'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 1–19. [Online]. Available: http://dl.acm.org/citation.cfm?id=1791834.1791836

[21] Dwork and Cynthia, "A firm foundation for private data analysis," *Commun. ACM*, vol. 54, no. 1, pp. 86–95, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1866739.1866758

[22] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati, "k-Anonymity," in *Secure Data Management in Decentralized Systems*, T. Yu and S. Jajodia, Eds. Springer-Verlag, 2007.

[23] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002. [Online]. Available: http://dx.doi.org/10.1142/S0218488502001648