Intrusion detection using Cloud computing

RPS Bedi

Deputy Registrar, PTU Jalandhar

Abstract:

Decades of research in the field of networking, web and software services has led to development of Cloud Computing. Cloud computing is an end-user friendly utility computing service. Cloud computing endeavors service oriented architecture, lessens information technology aloft for the end-user, provides great extensibility and lowers the total cost of ownership. Relying on the numerous resources provided by the cloud, distinct layers can be formalized as "Infrastructure as a Service" (IaaS), "Platform as Service" (PaaS), "Software as a Service" (SaaS), through which applications can be conveyed to the clients over the web. SaaS has come up as an indispensable service for the technology vendors to endeavor a wide variety of application services, application products and on demand services to their clients as per their usage and requirements over the web. Despite of all the benefits arising out of cloud computing, the distributed denial of service attack (DDoS) poses a threat to the utility of the cloud computing for future.

1. Introduction

Intrusion detection (Crothers, 2002) is the process of identifying and responding to suspicious activities targeted at computing and communication resources. An Intrusion detection system monitors and collects data from a target system that should be protected, processes and correlates the gathered information, and initiates responses when evidence of an intrusion is detected. Depending on their source of input, IDSs can be classified in to network based and host based systems. Network based intrusion detection systems (NIDs) collect input data by monitoring network traffic. Host based intrusion detection system collects events from monitored hosts. Host based IDS use information provided by the operating system to identify attacks. This information can be of different granularity and level of abstraction. However it usually relates to low level system operations such as system operations such as system calls, file system modifications and user logons. Because these operations represent a low level event stream, they usually contain reliable information and are difficult to tamper with, unless the system is compromised at the kernel level.

Cloud Computing has come up as a key service of the utility computing. It is built on decade of research in the field of networking, web and software services. In this thesis, implementation of intrusion detection with the help of cloud computing methods is done. Transformation of Computing to a model consisting of services that are customized and delivered in the same manner as traditional utilities like water, gas, electricity and telephony. In this type of model, based on their requirements users access services without regard to the place where the services are hosted or how they are being offered. Many computing systems have

promised to deliver this utility computing vision and having grid computing, cluster computing and most recently is Cloud computing.

These days, it is common to access content across the Internet infrastructure. This infrastructure has data centers that are monitored and maintained by content providers throughout. An extension of this paradigm is Cloud computing , wherein the capabilities of business utilities are exposed as useful services that can be accessed over a network. By charging consumers for accessing these services, the Cloud service providers are given incentives in terms of profits to be made.

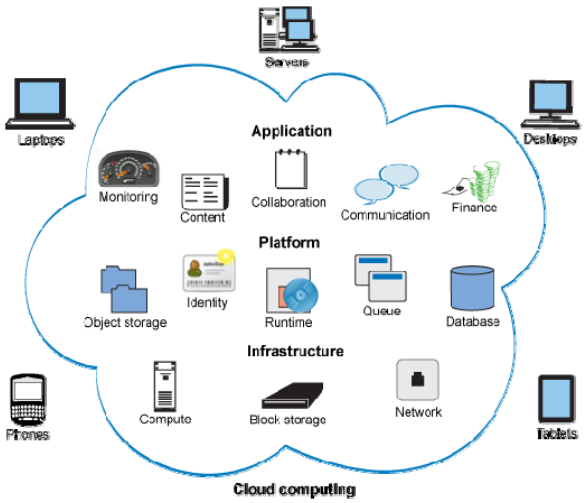

Figure 1.0 A Cloud Computing Environment.

Source Internet (free pictures)

Cloud Computing[1] as a utility, had been thought of a great revolution in the field of computing, which will upgrade IT industry in a big way, can make software a better attractive service and would shape the way IT hardware is conceptualized and purchased. No big efforts are required to the software professionals with imaginary thoughts for new Internet services in hardware to deploy their services or the human cost to operate it. So, the computing world is swiftly changing towards developing software for millions to use as a service, instead of running on their own computers.

Cloud Computing (Buyya et al. 2009) refers to the applications offered as services over the Internet as well as the hardware and systems software in the datacenters which provide these services**.** Cloud computing (Figure 1.0) is a model to allow requested network access , which is very convenient, to a shared group of configurable computing resources such as servers, storage, networks, applications and services which can be offered and released immediately with lesser service provider interaction or management effort .

2. Grid simulators

In the past decade, for delivering high-performance services to compute and data-intensive scientific applications grids have been evolved as the infrastructure. Several Grid simulators, such as GridSim , SimGrid, and GangSim have been proposed ,to support research and

development of new Grid components, policies, and middleware;. SimGrid is a generic framework for simulation of distributed applications on Grid platforms. Similarly, GangSim , a Grid simulation toolkit which helps for the modeling of Grid-based virtual organizations and resources. Whereas , for heterogeneous Grid resources ,GridSim is an event-driven simulation toolkit. Modeling of grid entities, users, machines, and network, including network traffic is supported by GridSim. Although all the above mentioned toolkits have capability to model and simulates the Grid application behaviors (execution, scheduling, allocation, and monitoring) in a distributed environment which consists of multiple Grid organizations but not even a single of it is able to support the infrastructure and application-level needs ,that arise from Cloud computing paradigm. In particular, there is a very little or no support in existing Grid simulation toolkits for modeling on-demand virtualization enabled resource and application management.

3. DDoS Impact on Clouds

Distributed Denial-of-service (DoS) attacks have been around for a long time. DDoS attacks usually take one of two forms, in the Cloud computing network arena (1) exploiting bugs in network clients or server applications, in an attempt to crash the application (2) flooding a network server with fake traffic, making it difficult or impossible for the server to receive and process legitimate traffic. Typically, instead overwriting critical information with the excess data, the former are carried out by using buffer overrun attacks' in which a network application is sent a huge quantity of data which it fails to grip properly. In general, some companies do not take security seriously enough and their systems are easily compromised and pose a threat not only to the companies themselves but also to anyone else targeted by a hacker through their systems.

4. Distributed Denial-of-Service Attack (DDoS)

A DDoS attack uses many computers to launch a coordinated DoS attack against one or more targets. The perpetrator is able to multiply the effectiveness of the DoS significantly by harnessing the resources of multiple unwitting accomplice computers, using client server technology, which serve as attack platforms. A DDoS attack is composed of four elements [6]. These are as follows:

1. The real attacker.
2. The handlers or master compromised hosts, capable of controlling multiple agents.
3. The attack daemon agents or zombie hosts, responsible for generating a stream of packets toward the intended victim.
4. A victim or target host.

5. Proposed Analytical Approach

As knowing the number of packet being malicious is a very uncertain problem, we have applied the probability theory to calculate that easily. Suppose the number of packets arriving at the server with a poisson's distribution ‘λ’.

Let us suppose further

p = probability of a packet being malicious

q *or* $1-p$ = probability of a packet being non-malicious

or legitimate

Now also suppose the packets are being arriving at server end with a Poisson's distribution 'λ'.

m = no of malicious packets

l = no of non-malicious or legitimate packets.

M = total no of packets arrived with Poisson's distribution 'λ'.

Now *Conditional Probability of each packet being malicious, under legitimate packets is

$$P(m, l) = P(m+l, m) \cdot P(m+l) \quad \text{.....(1)}$$

*(Note : Conditional Probability of occurrence of an event A ,once event E has occurred is given by $P(A/E) = P(A \cap E)/P(E)$)

Where $P(m+l, m)$ is the Probability of exactly occurrence of *'m'* success and is given by **Binomial Experiment as

$$P(m+l, m) = {}^{m+l}C_m \, p^m q^l$$

$$P(m+l, m) = m+l\,!\,/\,m\,!\,l\,!\; p^m q^l \quad \text{....(2)}$$

Further form Possion's Distribution

$$P(m+l) = e^{-\lambda} \lambda^{m+l} / m+l\,! \quad \text{.....(3)}$$

From equation (2) and (3) , we can rewrite equation (1) as :

$$P(m, l) = P(m+l, m) \cdot P(m+l)$$

$$= m+l\,!\,/\,m\,!\,l\,!\; p^m q^l \cdot e^{-\lambda} \lambda^{m+l} / m+l\,!$$

$$= m+l\,!\,/\,m\,!\,l\,!\; p^m (1-p)^l \cdot e^{-\lambda} \lambda^{m+l} / m+l\,! \quad (\text{replacing } q = 1-p)$$

$$= p^m (1-p)^l \left(e^{-\lambda p} e^{-\lambda(1-p)} \lambda^m \lambda^l / m\,!\,l\,! \right.$$

$$\mathbf{P(m, l) = e^{-\lambda p} (p^m \lambda^m) / m\,! \cdot e^{-\lambda(1-p)} (1-p)^l \lambda^l / l\,!} \quad \text{.....(4)}$$

Now from equation 4, the Joint probability of malicious packets in total traffic is given by

$$P(M = m) = \sum_{m=0}^{\infty} e^{-\lambda p}(p^m \lambda^m) / m ! \quad \text{.....(5)}$$

*(**Note : Probability of exactly 'k' success ,in Bionomical Experiment B (n.p) is given by* $P(n,k) = {}^{n}C_k\, p^k\, q^{n-k}$ *)*

From the above equation (5) it is interpreted that,

a) By setting the value of Joint Probability to 1 means that the intend of the attacker is to surpass the total channel bandwidth capacity by transfer as much as malicious packets as possible.
b) The value of Poisson's distribution 'λ' gives the rate of arrival of packets.
c) And approximate probability of malicious packet *'p*'.

Using above parameters we can find the value of 'm', the number of packets that are being malicious in the traffic.

The inspection algorithm calculates the source IP address and the final TTL value from each IP packet and then set the initial TTL value and subtracts the final TTL value for calculating the hop-count. The source IP address gives the index into the table to recover the correct hop-count for this IP address. If the calculated hop-count is same as the stored hop-count, the packet has been valid otherwise; the packet is likely to be spoofed. It is observed that a spoofed IP address may happen to have the same hop-count as the one from a zombie to the victim. In this scenario, HCF will not be capable to recognize the spoofed packet. But with a restricted range of hop-count values, HCF is extremely successful in calculating spoofed IP addresses.

Step 1: For each packet count the number of hops as H_{count} // By Hop Counter or Simple Inspection
Step 2: Retrieve the stored Hop count index as H_{stored}
Step 3: For each packet
if (H_{count} != H_{stored})
then 'discard the packet' // Packet is malicious
else
'allow the packet' // Packet is legitimate
Step 6 : end if

6. Effectiveness of HCIM Algorithm

The foundation behind hop-count Inspection is that a large amount spoofed IP packets, when arriving at victims, do not take hop-count values that are reliable with genuine IP packets from the sources that have been spoofed. Hop-Count Filtering (HCF) design a precise HCI mapping table, while using a reasonable quantity of storage, by adding address prefixes based on hop-

count. To detain hop-count changes under dynamic network conditions, a safe update system is devised for the HCI mapping table that prevents pollution by HCF aware attackers.

.

7. Proposed Algorithm (HCI- MPR)

As far as the HCIM is concerned the approach is not very effective in case of large data rate due to buffer overflow, secondary the approach needs network as well as administrative support. The detection rate of HCIM will reduced significantly when data rate is very high, so we are going to purpose a Algorithm named Hop Count Inspection with Malicious Probability Rate (HCI- MPR*)* which is based upon the results of the proposed Analytical Approach (3.1).
In our approach we are calculating the number of malicious packets from a given number of packets using a analytical approach (3.1). We first sample the number of packets on the basis of arrival at the server per a time unit. By taking that number of packets, the average arrival rate of the packets and the error probability of a packet we calculate the number of packets being malicious. Then we apply the simple HCIM (Hop-Count Inspection Method) Algorithm (3.5) to filter out first that many numbers of packets which was found out using the probabilistic approach. When we will reach at the exact number of packets, we simply release all the rest packets towards the server assuming that these are not malicious. It may also happen that we may lose some malicious packets being undetected but we save the computational time in a great extent than the actual HCIM Algorithm.

8. Proposed Algorithm

Hop Count Inspection with Malicious Probability Rate (HCI-MPR)

Step 1: For given value of 'λ' and 'p' calculate *'m'* (5) , no of malicious packets such that

$$P(M = m) = 1$$

(Joint probability of malicious packets)

Step 2: Initialize count = 1

Step 3: For each value of count =1 to m

Extract final value of TTL (Time to Live) as T_f

Investigate the initial value of TTL as T_i

Compute Hop count $H_c = T_f - T_i$

Retrieve the stored Hop count index as H_s

For each packet

if (H_c != H_s)

then 'discard the packet'
// Packet is malicious

else

'allow the packet'
// Packet is legitimate

Step 4: Increment count as count ++

Step 5: Repeat step 3 until count < = m.

Step 6: if count > m exit .

This algorithm is proposed to implement our approach at attacked end. Here initial input to the algorithm are the rate of arrival of the packets at the server, the error probability of each packet arrived at the server, and the number of packet was taken as a sample. Then these information were used to calculate the number of packets being malicious.. This value is input to the HCIM section. In HCIM section the packet will be checked whether the packet is 'spoofed′ or not using the discussed TTL value of each packet. Depending on the result of HCIM, the packet will either be allowed or dropped by the algorithm. If the packet will be found spoofed then the counter value will be incremented and the packet will be dropped. The counter value is maintained because the counter will run up to 'm' so that the first spoofed 'm' packets will be dropped and rest of the sampled packet will be considered as non-malicious and allowed to the server. Each time a spoofed packet will be identified the count value will be incremented. Hence by using this approach we are saving the computational time which was not saved in classical HCIM mechanism. In classical HCIM method the detection rate will also decrease when data rate increases.

9. Conclusion

In this our works focus on Distributed Denial-of-Service (DDoS) attack which is considered one of the harmful attacks in the Intrusion Detection. In this paper we have designed a new approach for intrusion detection using Cloud Computing, which is emerging as a key technology of the

future this attack had made the things crucial for the organizations which are providing their key service to the customers specially under Software as a Service (SaaS). This attack will harm the network within no time or without any prior knowledge which is providing servicers under SaaS. As this attack is very much harmful, so many defense mechanisms were developed to mitigate the attack to make the systems free form harm. The attack strategies are also of different types. They may need the help of network services or may not need to mitigate the attack. Till now so many network supported solution were proposed but proposals without the support of network were very rare. Traditionally HCIM (Hop count Inspection Method) method was there to mitigate the attack.

References:

1) Buyya Rajkumar, Yeoa Chee Shin, Venugopal Srikumar, Broberg James and Brandic Ivona (2009, pp 599-616) *"Cloud computing and emerging IT platforms : Vision, hype and reality for delivering computing as the 5thutility"* Future Generation Computer Systems Elsevier, Available online 11 December 2008.
2) Jensen Meiko, Schwenk Jorg, Gruschka Nils and Iacono Luigi Lo (2009) *"On Technical Security Issues in Cloud Computing"* IEEE International Conference on Cloud Computing.
3) Armbrust Michael, Fox Armando, Griffith Rean, Joseph Anthony D, Katz Randy H, Konwinski Andrew, Lee Gunho, Patterson David A, Rabkin Ariel, Stoica Ion and Zaharia Matei (2009) *"Above the Clouds: A Berkeley View of Cloud Computing"* Technical Report No. UCB/EECS-2009-28.
4) Greengard Samuel (2009) "Top 10 Trends in IT for 2009" , Baseline Online Magazine , published in CSI vol. no 33 , no 10 .
5) Kandukuri Balachandra Reddy, Paturi V Ramakrishna and Dr. Rakshit Atanu (2009) "*Cloud Security Issues*" IEEE International Conference on Services Computing.
6) Dikaiakos Marios D, Pallis George, Katsaros Dimitrios, Mehra Pankaj and Vakali Athena (2009) "CloudComputing-*Distributed Internet Computing for IT and Scientific Research*" IEEE Internet Computing .
7) Wang Cong, Wang Qian, Ren Kui and Lou Wenjing (2009) "*Ensuring Data Storage Security in Cloud Computing"* IEEE.
8) Xiong Kaiqi and Perros Harry (2009) "*Service Performance and Analysis in Cloud Computing*" Congress on Services – I, IEEE.
9) Buyya Rajkumar, Ranjan Rajiv and Calheiros Rodrigo N (2009) *"Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities"* IEEE.
10) Viega John (2009) "*Cloud Computing and the Common Man*" Computer , Published by the IEEE Computer Society .
11) Kumar Arun Raj and Selvakumar P. and S (2009) "*Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment - A Survey on DDoS Attack Tools and Traceback Mechanisms*" 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India.

12) Jack TIMOFTE, "Intrusion Detection using Open Source Tools" Revista Informatica Economică VOL.2 (46), pages 75-79, 2008.
13) Rui Santos, "Intrusion Prevention with L7-Filter" GSEC Gold Certification August 2008.
14) Renuka Prasad.B, Dr.Annamma Abraham, Chandan. C, Prabhanjan.A, and AjayBilotia "Information Extraction for Offline Traffic" IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.9, pages 309-315, September 2008
15) U. Aickelin, J. Twycross and T. Hesketh-Roberts, "Rule Generalisation in Intrusion Detection Systems using SNORT" International Journal of Electronic Security and Digital Forensics (IJESDF), Vol. 2, pp.20–26, 2008.
16) technet.microsoft.com."http://technet.microsoft.com/en-s/library/dd632948.aspx." Retrieved 2009-09-10.
17) Intrusion detection system "http://en.wikipedia.org/wiki/Intrusion detection system", cited October 2009.
18) OSI model "http://en.wikipedia.org/wiki/OSI model", cited August 2009.
19) Whitman, Michael, and Herbert Mattord. Principles of Information Security. Canada: Thomson, 2009. Pages 290 & 301
20) Pereira J. P., "Comparison of Firewall, Intrusion Prevention and Antivirus Technologies" accessed 2009.
21) Ceilyn Boyd John Zao David Mankins, Rajesh Krishnan and Michael Frentz. Mitigating distributed denial of service attacks with dynamic resource pricing. *Proceedings of 17th Annual Conference on Computer Security Applications , 2001. ACSAC 2001*, pages 411-421, 2001.
22) Xinzhou Qin Wenke Lee Ravi K. Prasanth B. Ravichandran Joao B. D. Cabr- era, Lundy Lewis and Ramon K. Mehra. Proactive detection of distributed denial of service attacks using mib tra±c variables - a feasibility study. *Integrated Network Management Proceedings*, pages 609-622, 2001.
23) John C. S. Lui David K. Yau and Feng Liang. Defending against distributed denial of service attacks with max-min fair server-centric router throttles. *2002 Tenth IEEE International Workshop on Quality of Service*, pages 35-44, 2002.
24) Guangsen Zhang and Manish Parashar. Cooperative defence against ddos attacks. *Department of Electrical and Computer Engineering, The State University of New Jersey*, February 2006.
25) S. Specht and R. Lee. Taxonomies of distributed denial of service networks, attacks, tools, and countermeasures. *Technical Report CE-L2003-03*, 164, May 2003.