Security Scheme for Protecting Cloud Computing Services Against Bursty DDoS Attacks

[1]Aws Naser , [2]Mohamad Fadli Zolkipli, Mazlina Abdul majid Mohamad. [3]Nusrat Ullah Khan[4]

[1,2,3]Fakulti Sistem Komputer & Kejuruteraan Perisian, Universiti Malaysia Pahang,
Lebuhraya Tun Razak, 26300 Gambang, Kuantan, Pahang Darul Makmur

Abstract

In cloud computing, data and applications are maintained on remote servers and accessed via the Internet. Virtualised resources such as dynamic servers are operated through the Internet, which increases the economic benefits accrued to customers from software. Cloud computing reduces customer concerns about software licenses, hardware, and overall system maintenance. Connections between web services are typically enabled using the simple object access protocol (SOAP), and extensible markup language (XML) or hypertext transport protocol (HTTP) is used to construct SOAP messages. Denial of service (DoS) and distributed DoS (DDoS) are two major problems affecting cloud computing services, and it is a challenge to resolve them completely. The identity of the perpetrators of these acts is usually difficult to ascertain especially when the attacks are carried out using spoofed IP addresses. Consequently, differentiating genuine packets from the packets sent by hackers is difficult. The addresses are spoofed with the intention of causing harm to cloud service provider communication channels. Distinguishing legitimate messages from illegitimate messages is an important step towards solving the problem of DDoS attacks. Modulo and CLASSIE methods effectively detect and reduce spoofing attacks using unique rulesets. In this paper, we propose using modulo packet marking and a method called reconstruct and drop (RAD) to differentiate and discard malicious packets. The proposed method improves the detection and filtering of DDoS attacks. Further, the results of comparisons conducted indicate that the proposed method requires fewer bits than Huffman code and its performance is better than that of cloud protector.

Keywords: *DDoS, reconstruct and drop (RAD), Huffman code*

1. Introduction

Cloud computing is a new technique that allows virtualised resources and services, such as infrastructure, platform, and software, from one or more physical servers to be simultaneously shared by many users over the Internet. In cloud computing, clients are able to access and utilise available resources and pay based on their demand[1]. The National Institute of Standards and Technology (NIST) in the United States defines cloud computing as a model that is convenient, ubiquitous, has an on-demand network and enables sharing of computing resources in a configurable manner[2]. The computing resources that are most frequently utilised are servers, networks, services, and applications, which can be released and provisioned with minimal interaction from service providers.

1.1.Cloud hallmarks

The fundamental identifiable features of cloud computing include rapid elasticity, resource pooling, on-demand self-service, and broadband network access[3]. At present, the categories of clouds include public, private, and community. Public clouds are available for public use, whereas private clouds are established in private organisations, where they are used only by the employees and customers of those organisations. Community clouds are used by groups of people who have the same objectives. Hybrid models that combine two or more of these categories also exist.

Cloud services include platform as a service (PaaS), software as a service (SaaS), and infrastructure as a service (IaaS) and are all delivered by the above models[4]. Providers allow clients to offload costly applications such as Customer relationship management (CRM) and Enterprise resource planning (ERP) and run them at the cost of the provider[5]. The shared resources provided include network bandwidth, operating system, and database.

1.2. Issue of security

The key concern when considering a cloud server is its ability to remain trustworthy. Currently, most organisations are attempting to offload both sensitive and insensitive data as they try to obtain sufficient useful data for analysis. Pay as you use is the fundamental basis on which cloud servers are operated. When numerous requests are sent through to servers as a result of denial of service (DoS) attacks, the recipient is forced to simultaneously process many more requests than they are able[6]. This leads to requests by legitimate users of the service being unfulfilled because the server is busy responding to the malicious requests sent by the DoS. The situation is compounded when multiple malicious requests are simultaneously coming from multiple sources, leading to a situation referred as distributed DoS (DDoS).

Attackers utilise various tools, such as Trinoo[7], Agobot , and Mstream [8], to carry out DDoS attacks. Most attackers tend to use less complicated attack tools based on the web, such as HTTP-based DoS (H-DoS) and XML-based DoS (X-DoS), which are simple to implement but difficult to counter[6].

The remainder of this paper is organised as follows: Section 2 reviews work related to cloud computing and DDoS. Section 3 presents our proposed approach. Section 4 discusses the results obtained. Finally, Section 5 concludes and outlines plans for future work.

2. Literature review

Prevention of legitimate access to specified data is achieved through introduction of DoS attackers. Through the Internet, attackers flood the connection of the victim, making it difficult for the user to access legitimate packets. In the recent past, DoS attacks have become increasingly common. This can be attributed to their ease of execution and the challenge tracing the source poses.

Tracing the source of a DoS attack is challenging to the users because the only method of detecting them is by identifying packets, and the packets are easily forged. A related study provided a solution comprising determination of the path followed by the traversed packet over the Internet through a process known as problem trace back[9-13]. This traceback method is based on reconstruction of polynomial and algebra techniques based on coding and learning theory, which provide reliable methods for reconstruction and transmission.

Deterministic packet marking (DPM) is an IP traceback approach proposed by Belenky[14]. In the proposed method, the packets are marked, with a one-bit reserved flag (RF) and an 18-bit ID field that are used together with an IP reader. The router interface closest to the packet source performs the marking. Only the information transferred with the DPM mark is then relied on for handling DDoS attacks. DPM marking thus aids the ingress of legitimate data in terms of continuous bit transfer. Additional information allows determination of segments of ingress address, which is important for understanding the specific destination. Users maintain a table that matches the ingress and source addresses that is maintained over time. A data structure referred to as a reconstruction table (rectbl) is utilised in the reconstruction procedure. On the arrival of ingress segments at their destination, there is a corresponding ingress address available to the identified user.

Xiang [15] proposed a method called flexible deterministic packet marking (FDPM), which acts as a system of defence with the ability to identify the real sources of packets that carry out attacks through network transverse. The IP header provides the different bits that are used by the FDPM method. The marks are flexible and are of varying lengths, according to the network protocols being used. They are typically referred to as length strategy flexible mark. FDPM flexibility is based on two main folds. The first is based on network protocols used in different networks. FDPM adaptability in relation to networks that are heterogeneous is one of the main characteristics that distinguish it. Secondly, FDPM is able to adjust and adapt to marking rate through the process of continuous marking. This characteristic is important in that it prevents overload problems that would be caused by router traceback. The characteristics have been utilised to enhance attack filtering and increases traceability of DDoS.

Hoi and Dai [16] presented a marking scheme which consist of traceback algorithms and markings that enable the router to mark links associated with the path that the packets followed. Traffic distribution guides the distribution of the links, and presents them in the form of Huffman codes. The marking made by a certain router does not belong to that router but to another router that sent the data to the identified router. The packet of data is sent according to a link table that allows identification of routers and their specific connectivity. Routers are guided by Huffman codewords, which present several links to other related routers at the router to which the packet has arrived.

Information is stored in messages formed and related to the marking field through codewords that are linked to Huffman. The router stores the identified data in the marked field, which are connected to the local memory of the user, which it clears on appending the codeword. The link of the stored message is later retrieved through the message digest that is directed to traceback the IP address. The traceback is accomplished via probabilistic markings, which require relatively less memory compared to different methods of logging, which are mainly associated with DDoS.

An IP traceback scheme that uses intelligent decision prototype (IDP), a machine learning technique, was proposed by Chonka et al. [17]. The two schemes outlined above, DPM and probabilistic packet marketing (PPM), can use IDP to identify DDoS attacks. IDP is a supervised machine learning application that comprises two parts. The first part is pre-marked decision (PMD) which, like DPM, is found on the boundaries of routers. Packets are relayed to the router that follows or host in cases of legitimate traffic. However, the packet is sent for marking when PMD realizes that the signs shown by the packet are not legitimate. The second part of IDP comprises two sections: One section reconstructs the path going back to the attacker's source, while the other section deals with the concrete attack packet by employing a machine learning method known as reconstruct and drop (RAD). This helps to reduce the marked packets, thus enhancing the efficiency and effectiveness of the system in locating the source of the attack.

An architectural standard and regulation called service oriented architecture (SOA) is used in the building of infrastructures; it eases the interaction between consumers and providers through services covering domains of ownership and technology. The service oriented traceback architecture (SOTA) is a new related approach proposed by Chonka [18] that provides a good framework for identifying the origin of an attack. In identifying the true source of DDoS, the SOA approach is used by SOTA as the main traceback methodology. The DPM packet-marking strategy is also utilised by SOTA. In this context, the DPM methodology is employed to place a service oriented traceback mark (SOTM) in web service messages. SOTA was further extended by Chonka et al. [19]to enable it to prevent DDoS attacks on web services. SOTA primarily identifies the true identity of fabricated messages, because attackers hide their identity in order that the defence system is barred from accessing the fabricated message. In order to function effectively the source of the attack should be close to SOTA. An inbound SOAP message is tagged with a SOAP header upon arrival at the router. The header is used to traverse the network back to the source of the attack. Chonka applied the framework to the open grid service architecture (OGSA) and provides a filter for defence known as XML Detector (XDetector), which defends it effectively by being evenly distributed in the grid. To enable detection and filtering of messages by XML-based DoS (X-DoS) attacks, a back propagation neural network is used to train the XML-based detector. For high efficiency and protection, the XDetector is placed in front of the web server.

Chonka et al. [17] proposed a cloud-based traceback service architecture for the prevention of DDoS. In the proposed scheme, the origin of attacks is traced by cloud traceback (CTB). CTB also uses a cloud protector (XDetector), which filters attack traffic. In an attack scenario, when a web service is requested by an attack client from CTB, the request is passed to the web server, which formulates a SOAP message on the basis of the description of the service by the attack client. SOTA then places SOTM within the header of the SOAP message. The SOAP message is relayed to the web server on the placing of the CTBM. The victim of an attack then asks for reconstruction to obtain the mark and transfer information about the source of the message. Reconstruction filters the attack traffic and also identifies and filters the source of the attack in a short time

3. Methodology

3.1. Proposed scheme

In a typical DDoS attack, a machine is sent such a flood of messages that it can only manage to handle a few requests at a time, or the system ends up collapsing. Some types of DDoS attacks, such as XML and HTTP DoS attacks, can crash crowd web services. A combination of these two attacks is called an HX-DoS attack. In a DDoS attack scenario, the attacker may manage to comprise an individual who has access to an account on a server of the cloud service provider prior to the attack. This enables the attacker to connect directly through the system. The DoS attack program is installed by the attacker at the user's end and is initiated. One of the methods used to differentiate attacks is the Intrusion Detection System (IDS) where a decision tree classification system known as CLASSIE is used.

CLASSIE is typically located one hop away from the host. Its ruleset has overtime been developed to recognize known DDoS messages. With known DDoS attacks such as XML payload overload and XML injection, it is possible to train and test CLASSIE to recognize the known attributes. When CLASSIE detects a DDoS message, it drops the packet matching the ruleset. Following examination by CLASSIE, the packets are then subjected to marking. Figure 1 gives a conceptual overview of our proposed approach. We propose a modulo packet-marking algorithm that marks packets with router information as they move through the network. Following a traceback request, the path navigated by the packets is reconstructed using reverse modulo.

The marking is carried out on both core and edge routers. When an incoming packet is to be marked by an edge router, the code to be marked is fetched if, it matches the physical address of the host from the lookup table, and encoded into the packet. One bit is required by the edge router to indicate whether the packet has been marked and a few bits are used for marking code. A lookup table known as a MACtoID table is maintained. The table contains the physical addresses of the hosts that are attached to the network along with their corresponding equivalent numeric code. The algorithm that performs the actual marking at the edge router uses the following steps: Step 1: For each packet, use the sender's physical address to locate the code to be marked in the MACtoID table. Step 2: Set the marked field. Step 3: Stamp the code into the marked field. Step 4: Forward the packet to the next router.

A core router can only mark after the edge router has marked the packet; otherwise, the packets are simply forwarded. A table with the physical addresses of all the input addresses of the hardware and the link numbers that are assigned to all the interfaces, known as a MAC to Interface table, is maintained by the core router. The algorithm used to conduct the marking at the core router comprises the following steps: Step 1: For each packet, after the marked field has been set, use the MAC to Interface table to locate the link number for the inbound interface on which the packet arrived. Step 2: Calculate the new marking information. Step 3: Forward the packet to the next router. The table is consulted when the router decides to mark so as to locate the link number that has been assigned to the inbound interface. The modulo technique for marking is used by the core router and the Eq. 1 shows its calculation. RAD is constructed from IDP and is located one hop back from the victim.

$$\text{New marking information} = \\ \text{current marking information} \times \text{number of interfaces on router} + \text{link number} \qquad (1).$$

In general, the same path (shortest path) is followed by the host across the router to send packets to destinations. After the marking value and stored value matches, the packet is forwarded to the respective hosts. During an attack in which the IP address of a host is spoofed by another host, the marking value of the packet differs from the value that is stored in the RAD. The reason is that when it comes to marking, CLASSIE utilises the MAC address and not the IP address. Therefore, the packets are left at the side of the victim and RAD asks for a traceback. When a victim is being attacked, a traceback request containing the marking information of the packet that needs to be traced is issued to the closest router responsible for delivering the packet. The reverse modulo is then used by the upstream router to locate the inbound interface of the offending packet with the use of marking information located in the traceback request. The hardware address table located at the inbound interface is consulted and the router locates the previous upstream router connected to that interface.

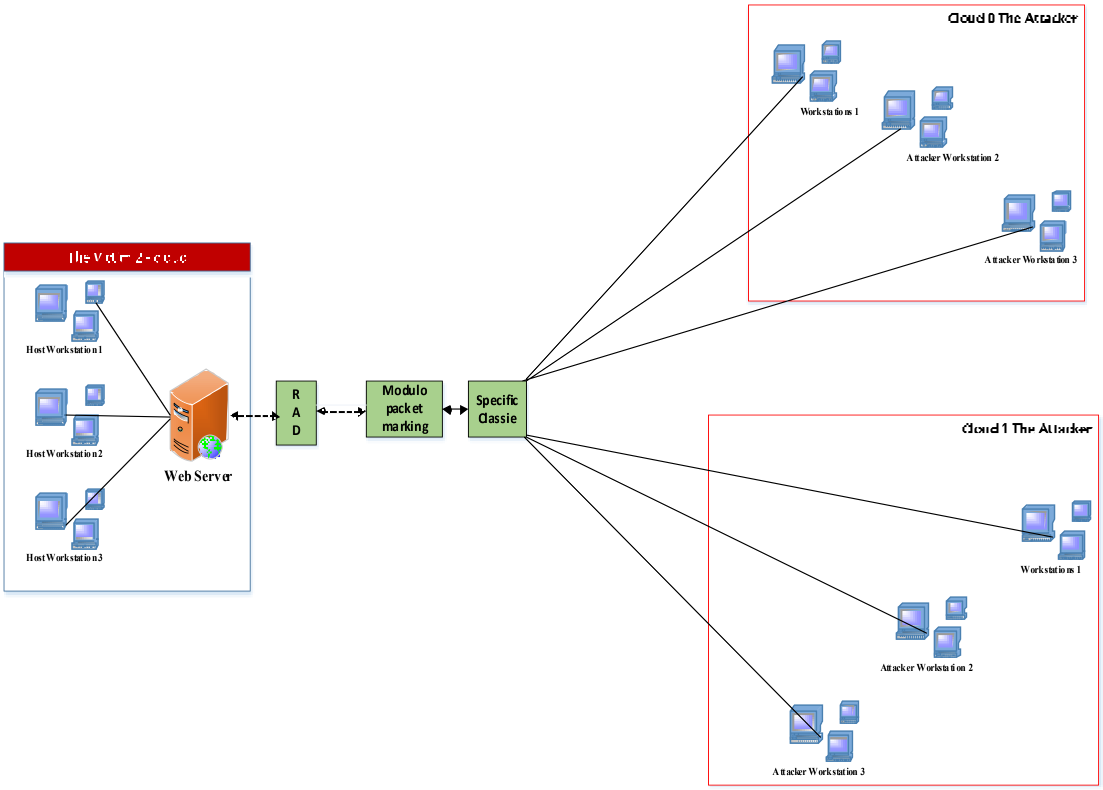

Figure 1. Conceptual overview of our proposed approach

Subsequently, the upstream router becomes the current router and the traceback procedure is performed repeatedly until the edge router of the sending host is reached. Consequently, the victim is able to find the routers that were crossed by the attack packet and a request is sent to the edge router to locate the physical address of the node that the attack packet originated from. RAD functions by observing the incoming messages and decides if it should allow the messages through or drop them. The spoofing attack is thus prevented after it finds the true origin of the packet.

4. Results and discussion

False alarm rate and detection rate (DR) are two of the most crucial parameters in detection and filtering during DDoS attacks. The DR of an attack traffic that has been trained and tested by CLASSIE equals true positive (TP). TP is defined as a system getting an alert when an attack occurs. False positive is defined as an alert being received by a system even though an attack has not taken place. True negative is defined as no alert being issued when there is no attack. True negative is usually the default since there are no intruder attacks and therefore no alerts from the alarm are being received.

When the attacks from intruders occur without being noticed by the system, this is regarded as false negative (FP) is defined.

A comparison of the mean length of the code required by our proposed scheme to the average length required by the Huffman code, indicate that the Huffman code requires more bits than our proposed modulo packet marking. Further, analysis of the performance of the proposed scheme to detect and trace different numbers of attack packets indicate that the module packet-marking performance is above that of the cloud protector.

5. Conclusion and future work

XML-based and HTTP DoS attacks are among some of the most serious threats affecting cloud computing. A packet-based marking approach can appropriately respond and detect such threats, filter them from the attacker side, and drop them. This action helps to reduce instances of DoS attacks, which further reduces the rate of attacks. In this paper, we proposed using RAD in place of cloud protector to enhance the ability to detect attacks and thus introduce modulo and CLASSIE marking on the source side. Our proposed scheme improves the rate of detection of DDoS attacks that are known to affect the reliability of transferred data. It also helps to improve the FP rate through improved filtering and detection of DDoS attacks. In future work, we plan to extend our proposed system by combining it with defensive source end systems that facilitate detection based on MAC spoofing.

6. References

[1] R. Latif, H. Abbas, S. Assar, and Q. Ali, "Cloud Computing Risk Assessment: A Systematic Literature Review," *Future Information Technology*, pp. 285-295: Springer, 2014.

[2] P. Mell, and T. Grance, “The NIST definition of cloud computing (draft),” *NIST special publication,* vol. 800, no. 145, pp. 7, 2011.

[3] K. Popovic, and Z. Hocenski, "Cloud computing security issues and challenges." pp. 344-349.

[4] A. Prasad, P. Green, and J. Heales, "On cloud computing service considerations for the small and medium enterprises."

[5] K. Yang, W. Zhao, J.-s. Zhang, and X.-l. Li, "Research of CRM/ERP Integrated Systems for New Materials SMEs with Scattered Customers." pp. 119-127.

[6] M. Behzadi, R. Mahmod, M. Barati, A. B. H. Abdullah, and M. Noura, “A New Framework for Classification of Distributed Denial of Service (DDOS) Attack in Cloud Computing by Machine Learning Techniques,” *Advanced Science Letters,* vol. 20, no. 1, pp. 175-178, 2014.

[7] M. Kaur, and S. Vashist, “A REVIEW OF THE DOS-DDOS ATTACKS AND THEIR PREVENTION MECHANISMS IN CLOUD.”

[8] V. Krylov, and K. Kravtsov, “DDoS Attack and Interception Resistance IP Fast Hopping Based Protocol,” *arXiv preprint arXiv:1403.7371*, 2014.

[9] S. Yu, "DDoS Attack and Defence in Cloud," *Distributed Denial of Service Attack and Defense*, pp. 77-93: Springer, 2014.

[10] S. N. Prabhu, and D. Shanthi, “A Survey on Anomaly Detection of Botnet in Network,” *International Journal,* vol. 2, no. 1, 2014.

[11] S. Sharma, G. Gupta, and P. Laxmi, “A Survey on Cloud Security Issues and Techniques,” *arXiv preprint arXiv:1403.5627*, 2014.

[12] I. M. Khalil, A. Khreishah, and M. Azeem, “Cloud computing security: a survey,” *Computers,* vol. 3, no. 1, pp. 1-35, 2014.

[13] Y. Wang, and R. Sun, “An IP-Traceback-based Packet Filtering Scheme for Eliminating DDoS Attacks,” *Journal of Networks,* vol. 9, no. 4, 2014.

[14] A. Belenky, and N. Ansari, “IP traceback with deterministic packet marking,” *IEEE Communications Letters,* vol. 7, no. 4, pp. 162-164, 2003.

[15] Y. Xiang, W. Zhou, and M. Guo, “Flexible deterministic packet marking: An ip traceback system to find the real source of attacks,” *Parallel and Distributed Systems, IEEE Transactions on,* vol. 20, no. 4, pp. 567-580, 2009.

[16] K. Choi, and H. Dai, "A marking scheme using Huffman codes for IP traceback." pp. 421-428.

[17] A. Chonka, Y. Xiang, W. Zhou, and A. Bonti, “Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks,” *Journal of Network and Computer Applications,* vol. 34, no. 4, pp. 1097-1107, 2011.

[18] A. Chonka, W. Zhou, and Y. Xiang, "Protecting web services with service oriented traceback architecture." pp. 706-711.

[19] A. Chonka, W. Zhou, and Y. Xiang, "Defending grid web services from xdos attacks by sota." pp. 1-6.