Volume 3, Issue 5, May-2016, pp. 369-375 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Defensive Cloud Service Providers Against Stealthy Denial of Service Strategy

[1]Mrs. P.SRILAKSHMI, [2] Mrs. N.SUJATHA

[1] Pursuing M.Tech(CSE)from Jagruti Institute of Engineering and Technology

[2] Associate Professor, Department of Computer Science and Engineering, Jagruti Institute of Engineering and Technology, Telangana State, India.

Abstract— Cloud Computing allows customers to access cloud resources and services. On-demand, self-service and pay-by-use business model are adapted for the cloud resource sharing process. Service level agreements (SLA) regulate the cost for the services that are provided for the customers. Cloud data centers are employed to share data values to the users. Denial-of-Service (DoS) attack is an attempt by attacker to prevent legitimate users from using resources. Distributed Denial of Service (DDoS) Attacks is generated in a "many to one" dimension. In DDoS attack model large number of compromised host are gathered to send useless service requests, packets at the same time .DoS and DDoS attacks initiates the service degradation, availability and cost problems under cloud service providers. Brute-force attacks are raised against through specific periodic, pulsing and low-rate traffic patterns. Rate-controlling, time-window, worst-case threshold and pattern-matching are adapted to discriminate the legitimate and attacker activities. Stealthy attack patterns are raised against applications running in the cloud. Slowly-Increasing-Polymorphic DDoS Attack Strategy (SIPDAS) can be applied to initiate application vulnerabilities. SIPDAS degrades the service provided by the target application server running in the cloud. Polymorphic attacks changes the message sequence at every successive infection to avoid signature detection process. Slowly-increasing polymorphic behavior induces enough overloads on the target system. XML-based DoS (XDoS) attacks to the web-based systems are applied as the testing environment for the attack detection process we describe both how to apply the proposed strategy, and its effects on the target system deployed in the cloud.

Keywords—Raincloud computing, erudite attacks strategy, stumpy-rate attacks, interference detection.

1. INTRODUCTION

Cloud Computing is an emerging paradigm that allows customers to obtain cloud resources and services according to an on-demand, self-service, and pay-by-use business model. Service Level Agreements (SLA) regulates the costs that the cloud customers have to pay for the provided Quality of Service (QoS) [1]. A side effect of such a model is that, it is prone to DoS and Distributed DoS (DDoS), which aim at reducing the service availability and performance by exhausting the resources of the service's host system (including memory, processing resources, and network bandwidth) [2]. Such attacks have special effects in the cloud due to the adopted pay-by-use business model. Specifically, in Cloud Computing also partial service degradation due to an attack has direct effect on the service costs, and not only on the performance and availability perceived by the customer. The delay of the cloud service provider to diagnose the causes of the service degradation (i.e., if it is due to either an attack or an overload) can be considered as a security vulnerability. It can be exploited by attackers that aim at exhausting the cloud resources (allocated to satisfy the negotiated QoS), And seriously degrading the QoS, as

happened to the Bit Bucket Cloud, which went down for 19h [3]. Therefore, the cloud management system has to implement specific countermeasures in order to avoid paying credits in case of accidental or deliberate intrusion that cause violations of QoS guarantees. Over the past decade, many efforts have been devoted to the detection of DDoS attacks in distributed systems. Security prevention mechanisms usually use approaches based on rate controlling, time window, worst-case threshold, and pattern matching methods to discriminate between the nominal system operation and malicious behaviors [4]. On the other hand, the attackers are aware of the presence of such protection mechanisms. They attempt to perform their activities in a "stealthy" fashion in order to elude the security mechanisms, by orchestrating and timing attack patterns that leverage specific weaknesses of target systems [5]. They are carried out by directing flows of legitimate service requests against a specific system at such a low-rate that would evade the DDoS detection mechanisms, and prolong the attack latency, i.e., the amount of time that the ongoing attack to the system has been undetected. We show that the features offered by the cloud provider, to ensure the SLA negotiated with the customer (including the load balancing and autoscaling mechanisms), can be maliciously exploited by the proposed stealthy attack, which slowly exhausts the resources provided by the cloud provider, and increases the costs incurred by the customer.

2. BACKGROUND AND RELATED WORK

Sophisticated DDoSattaks are defined as that category of attacks, which are tailored to hurt a specific weak point in the target system design, in order to conduct denial of service or just to significantly degrade the performance [12], [7]. The term stealthy has been used in [13] to identify sophisticated attacks that are specifically designed to keep the malicious behaviors virtually invisible to the detection mechanisms. The methods of launching sophisticated attacks can be categorized into two classes: job-content based and jobs arrival pattern-based. The former have been designed in order to achieve the worst-case complexity of O(n) elementary operations per submitted job, instead of the average case complexity of O(1) [14], [15], [16]. The jobs arrival pattern based attacks exploit the worst case traffic arrival pattern of requests that can be applied to the target system [7], [17]. In general, such sophisticated attacks are performed by sending low-rate traffic in order to be unnoticed by the DDoS detection mechanisms. Due to its high similarity to legitimate network traffic and much lower launching overhead than classic DDoSattack, this new assault type cannot be efficiently detected or prevented by existing network-based solutions [21], [22]. Therefore, in recent years, the target of DDoS attacks has shifted from network to application server resources and procedures. The attack takes advantage of the capacity to forecast the time at which the responses to incoming requests for a given service occur. This capability is used to schedule an intelligent pattern in such a way that the attacked server becomes busy the most time in processing of the malicious requests instead of those from legitimate users.

2.1 Cloud Resources Provisioning Cloud providers offer services to rent computation and storage capacity, in a way as transparent as possible, giving the impression of 'unlimited resource availability'. However, such resources are not free. Therefore, cloud providers allow customers to obtain and configure suitably the system capacity, as well as to quickly renegotiate such capacity as their requirements change, in order that the customerscanpay only for resources that they actually use. Several cloud providers offer the 'load balancing' service for automatically distributing the incoming application service requests across multiple instances, as well as the 'auto scaling' service for enabling consumers to closely follow the demand curve for their applications (reducing the need to acquire cloud resources in advance). In order to minimize the customer costs, the auto scaling ensures that the number of the application instances increases seamlessly during the

demand spikes (to maintain the contracted performance), and decreases automatically during the demand lulls.

3. HANDLING DENIAL OF SERVICE ATTACKS IN CLOUD

Cloud Computing is an emerging paradigm that allows customers to obtain cloud resources and services according to an on-demand, self-service, and pay-by use business model. Service level agreements (SLA) regulate the costs that the cloud customers have to pay for the provided quality of service (QoS) [1]. A side effect of such a model is that, it is prone to Denial of Service (DoS) and Distributed DoS (DDoS), which aim at reducing the service availability and performance by exhausting the resources of the service's host system. Such attacks have special effects in the cloud due to the adopted pay-by-use business model. Specifically, in cloud computing also partial service degradation due to an attack has direct effect on the service costs, and not only on the performance and availability perceived by the customer. The delay of the cloud service provider to diagnose the causes of the service degradation can be considered as security vulnerability. It can be exploited by attackers that aim at exhausting the cloud resources and seriously degrading the QoS, as happened to the BitBucket Cloud, which went down for 19h. Therefore, the cloud management system has to implement specific countermeasures in order to avoid paying credits in case of accidental or deliberate intrusion that cause violations of QoS guarantees. Over the past decade, many efforts have been devoted to the detection of DDoS attacks in distributed systems. Security prevention mechanisms usually use approaches based on rate controlling, time-window, worst-case threshold, and pattern-matching methods to discriminate between the nominal system operation and malicious behaviors. On the other hand, the attackers are aware of the presence of such protection mechanisms. They attempt to perform their activities in a "stealthy" fashion in order to elude the security mechanisms, by orchestrating and timing attack patterns that leverage specific weaknesses of target systems. They are carried out by directing flows of legitimate service requests against a specific system at such a low-rate that would evade the DDoS detection mechanisms, and prolong the attack latency, i.e., the amount of time that the ongoing attack to the system has been undetected. This paper presents a sophisticated strategy to orchestrate stealthy attack patterns against applications running in the cloud. Instead of aiming at making the service unavailable, the proposed strategy aims at exploiting the cloud flexibility, forcing the application to consume more resources than needed, affecting the cloud customer more on financial aspects than on the service availability. The attack pattern is orchestrated in order to evade, greatly delay the techniques proposed in the literature to detect low-rate attacks. It does not exhibit a periodic waveform typical of low-rate exhausting attacks [8]. In contrast with them, it is an iterative and incremental process. In particular, the attack potency is slowly enhanced by a patient attacker, in order to inflict significant financial losses, even if the attack pattern is performed in accordance to the maximum job size and arrival rate of the service requests allowed in the system. Using a simplified model empirically designed, we derive an expression for gradually increasing the potency of the attack, as a function of the reached service degradation. We show that the features offered by the cloud provider, to ensure the SLA negotiated with the customer can be maliciously exploited by the proposed. Stealthy attack, which slowly exhausts the resources provided by the cloud provider and increases the costs incurred by the customer. The proposed attack strategy, namely Slowly-Increasing-Polymorphic DDoS Attack Strategy (SIPDAS) can be applied to several kind of attacks, that leverage known application vulnerabilities, in order to degrade the service provided by the target application server running in the cloud. The term polymorphic is inspired to polymorphic attacks which change message sequence at every successive infection in order to evade signature detection mechanisms [9]. Even if the victim detects the SIPDAS attack, the attack

strategy can be reinitiate by using a different application vulnerability, or a different timing. In order to validate the stealthy characteristics of the proposed SIPDAS attack, we explore potential solutions proposed in the literature to detect sophisticated low-rate DDoS attacks. We show that the proposed slowly-increasing polymorphic behavior induces enough overload on the target system and evades, or however, delays greatly the detection methods. In order to explore the attack impact against an application deployed in a cloud environment, this paper focuses on one of the most serious threats to cloud computing, which comes from XMLbased DoS (XDoS) attacks to the web-based systems [10]. The experimental testbed is based on the mOSAIC framework, which offers both a 'Software Platform' that enables the execution of applications developed using the mOSAIC API, and a 'Cloud Agency', that acts as a provisioning system, brokering resources from a federation of cloud providers [11].

4. PROBLEM STATEMENT

Brute-force attacks are raised against through specific periodic, pulsing and low-rate traffic patterns. Rate-controlling, time-window, worst-case threshold and pattern-matching are adapted to discriminate the legitimate and attacker activities. Stealthy attack patterns are raised against applications running in the cloud. Slowly-Increasing- Polymorphic DDoS Attack Strategy (SIPDAS) can be applied to initiate application vulnerabilities. SIPDAS degrades the service provided by the target application server running in the cloud. Polymorphic attacks changes the message sequence at every successive infection to avoid signature detection process. Slowly-increasing polymorphic behavior induces enough overloads on the target system. XMLbased DoS (XDoS) attacks to the web-based systems are applied as the testing environment for the attack detection process. The following drawbacks are identified from the existing system. • SIPDAS based attack detection is not supported • Polymorphic behavior identification is not adapted • Application level vulnerability detection is low • Service degradation and resource consumption cost analysis is not performed

5. STEALTHY DOS ATTACKS ON CLOUD SERVICES

5.1. DoS Attacks Against Cloud Applications

In this section are presented several attack examples, which can be leveraged to implement the proposed SIPDAS attack pattern against a cloud application. In particular, we consider DDoS attacks that exploit application vulnerabilities, including: the Oversize Payload attack that exploits the high memory consumption of XML processing; the Oversized Cryptography that exploits the flexible usability of the security elements defined by the WSSecurity specification the Resource Exhaustion attacks use flows of messages that are correct regarding their message structure, but that are not properly correlated to any existing process instance on the target server and attacks that exploit the worst-case performance of the system, for example by achieving the worst case complexity of Hash table data structure, or by using complex queries that force to spend much CPU time or disk access time. In this paper, we use a Coercive Parsing attack as a case study, which represents one of the most serious threat for the cloud applications. It exploits the XML verbosity and the complex parsing process. In particular, the Deeply-Nested XML is a resource exhaustion attack, which exploits the XML message format by inserting a large number of nested XML tags in the message body. The goal is to force the XML parser within the application server, to exhaust the computational resources by processing a large number of deeply-nested XML tags.

5.2. Stealthy Attack Objectives

The system is aimed to defining the objectives that a sophisticated attacker would like to achieve, and the requirements the attack pattern has to satisfy to be stealth. Recall that, the purpose of the attack against cloud applications

is not to necessarily deny the service, but rather to inflict significant degradation in some aspect of the service, namely attack profit PA, in order to maximize the cloud resource consumption CA to process malicious requests. In order to elude the attack detection, different attacks that use low-rate traffic have been presented in the literature. Therefore, several works have proposed techniques to detect low-rate DDoS attacks, which monitor anomalies in the fluctuation of the incoming traffic through either a time or frequency-domain analysis. They assume that, the main anomaly can be incurred during a low- rate attack is that, the incoming service requests fluctuate in a more extreme manner during an attack. The abnormal fluctuation is a combined result of two different kinds of behaviors: (i) a periodic and impulse trend in the attack pattern, and (ii) the fast decline in the incoming traffic volume. Therefore, in order to perform the attack in stealthy fashion with respect to the proposed detection techniques, an attacker has to inject low-rate message flows $\phi Aj = [\phi j,1, \ldots, \phi j,m]$, that satisfy the following optimization problem:

5.3. Attack Approach

In order to implement SIPDAS-based attacks, the following components are involved:

- a Master that coordinates the attack;
- π Agents that perform the attack; and
- a Meter that evaluates the attack effects. The approach implemented by each Agent to perform a stealthy service degradation in the cloud computing. It has been specialized for an X-DoS attack. Specifically, the attack is performed by injecting polymorphic bursts of length T with an increasing intensity until the attack is either successful or detected. Each burst is formatted in such a way as to inflict a certain average level of load CR. In particular, we assume that CR is proportional to the attack intensity of the flow ΦAj during the period T. Therefore, denote I0 as the initial intensity of the attack, and assuming $\Delta CR = \Delta I$ as the increment of the attack intensity.

For each attack period, fixed the maximum number of nested tags (tagThreshold), the routine pickRandomTags(. . .) randomly returns the number of nested tags nT for each message. Based on nT , the routine compute Inter arrival Time uses a specific algorithm to compute the inter-arrival time for injecting the next message.

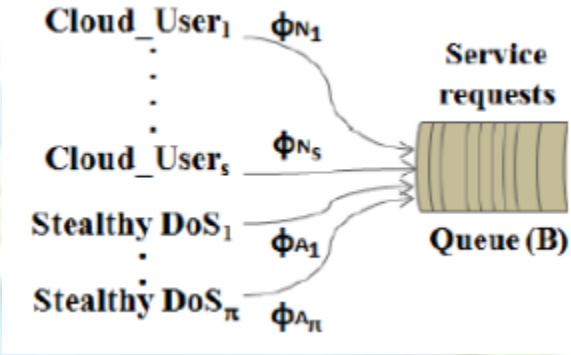

Fig 1. Attach approach

At the end of the period T, if the condition 'attack Successful' is false, the attack intensity is increased. If the condition 'attack Successful' is true, the attack intensity is maintained constant until either the attack is detected or the auto-scaling mechanism enabled in the cloud adds new cloud resources. The attack is performed until it is either detected, or the average message rate of the next burst to be injected is greater than dT. In this last case, the Agent notifies to the Master that the maximum average message rate is reached and continues to inject messages formatted according to the last level of load CR reached.

6. FURTIVE DOS DESCRIPTION AND MODELING

This section defines the characteristics that a DDoS attack against an application server running in the cloud should have to be stealth. quality of service provided to the user, we assume that the system performance under a DDoS attack is more degraded, as higher the average time to process the user service requests 3.2 Server Under Attack Model In order to assess the service degradation attributed to the attack, we define a synthetic representation of the

system under attack. We suppose that the system consists of a pool of distributed VMs provided by the cloud provider, on which the application instances run.

7. CONCLUSIONS

In this paper, we propose a strategy to implement stealthy attack patterns, which exhibit a slowly-increasing polymorphic behavior that can evade, or however, greatly delay the techniques proposed in the literature to detect low-rate attacks. Exploiting a vulnerability of the target application, a patient and intelligent attacker can orchestrate sophisticated flows of messages, indistinguishable from legitimate service requests. In particular, the proposed attack pattern, instead of aiming at making the service unavailable, it aims at exploiting the cloud flexibility, forcing the services to scale up and consume more resources than needed, affecting the cloud customer more on financial aspects than on the service availability. The system minimizes the application level vulnerabilities. Attack behavioral changes are automatically detected by the system.

REFERENCES

[1] M. C. Mont, K. McCorry, N. Papanikolaou, and S. Pearson, "Security and privacy governance in cloud computing via SLAS and a policy orchestration service," in Proc. 2nd Int. Conf. Cloud Comput. Serv. Sci., 2012, pp. 670–674.

[2] S. Malek and S. Salvatore, "Detecting masqueraders: A comparison of one-class bag-ofwords user behavior modeling techniques," in Proc. 2nd Int. Workshop Managing Insider Security Threats, Morioka, Iwate, Japan. Jun. 2010, pp. 3–13.

[3] A. S. Sodiya, O. Folorunso, S. A. Onashoga, and P. O. Ogundeyi, "An improved semi-global alignment algorithm for masquerade detection," Int. J. Netw. Security, vo1. 12, no. 3, pp. 211– 220, May 2011.

[4] Yongdong Wu, Zhigang Zhao, Feng Bao and Robert H. Deng, "Software Puzzle: A Countermeasure to Resource-Inflated Denial-of-Service Attacks", IEEE Transactions On Information Forensics And Security, Vol. 10, No. 1, January 2015

[5] Hisham A. Kholidy, Fabrizio Baiardi and Salim Hariri, "DDSGA: A Data-Driven SemiGlobal Alignment Approach for Detecting Masquerade Attacks", IEEE Transactions On Dependable And Secure Computing, Vol. 12, No. 2, March/April 2015

[6] Subrat Kumar Dash, K. S. Reddy, and K. A. Pujari, "Adaptive Naive Bayes method for masquerade detection", Security Commun. Netw., vol. 4, no. 4, pp. 410–417, 2011.

[7] Guojun Wang, Felix Musau, Song Guo and Muhammad Bashir Abdullahi, "Neighbor Similarity Trust against Sybil Attack in P2P E-Commerce", IEEE Transactions On Parallel And Distributed Systems, Vol. 26, No. 3, March 2015

[8] X. Xu, X. Guo, and S. Zhu, "A queuing analysis for low-rate DoS attacks against application servers," in Proc. IEEE Int. Conf. Wireless Commun., Netw. Inf. Security, 2010, pp. 500–504.

[9] L. Wang, Z. Li, Y. Chen, Z. Fu, and X. Li, "Thwarting zero-day polymorphic worms with network-level length-based signature generation," IEEE/ACM Trans. Netw., vol. 18, no. 1, pp. 53–66, Feb. 2010.

[10] A. Chonka, Y. Xiang, W. Zhou, and A. Bonti, "Cloud security defense to protect cloud computing against HTTP-DOS and XMLDoS attacks," J. Netw. Comput. Appl., vol. 34, no. 4, pp. 1097–1107, Jul. 2011.

[11] D. Petcu, C. Craciun, M. Neagul, S. Panica, B. Di Martino, S. Venticinque, M. Rak, and R. Aversa, "Architecturing a sky computing platform," in Proc. Int. Conf. Towards Serv.-Based Int., 2011, vol. 6569, pp. 1-13.

ABOUT THE AUTHORS

Mrs. P.SRILAKSHMI is pursuing M.Tech degree in, Computer Science and Engineering from Jagruti Institute of Engineering and Technology, Telangana State, India.

Mrs.N.SUJATHA is presently working as Associate Professor in, Department of computer science and engineering, Telangana State,India.She has published several research papers in both International and National conferences and Journals.

