International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4 | ISSN : 2456-3307

Minimizing The Denial of Service Attack By Using The Cloud Information Framework

G. Dhana Sekar[1], M.Vidhyasree[2], S. Radhika[2]
[1]Assistant Professor, Department of CSE, Sri Venkateswara College of Engineering And Technology, Chittoor, Andhra Pradesh, India
[2]Student, Department of CSE, Sri Venkateswara College of Engineering And Technology , Chittoor, Andhra Pradesh, India

ABSTRACT

Due to the close correlation with individual's physical features and status, the adoption of Cyber-Physical Social Systems (CPSSs) has been inevitably hindered by users' privacy concerns. The success of the Cloud Computing paradigm is owing to its on-demand, self-service, and pay-by-use nature. Attacks involve not solely the standard of the delivered service, but conjointly the service maintenance prices in terms of resource consumption. Specifically, the longer the detection delay is, the higher the prices to be incurred. Therefore, a selected attention has to be procured stealthy DoS attacks. They aim at minimizing their visibility, and at identical time, they will be as harmful as the brute-force attacks. They're subtle attacks tailored to leverage the worst-case performance of the target system through specific periodic, pulsing, and low-rate traffic patterns. In this paper, we have a tendency to propose a method to orchestrate stealthy attack patterns, that exhibit a slowly-increasing-intensity trend designed to bring down the most monetary value to the cloud customer, whereas respecting the work size and therefore the service arrival rate obligatory by the detection mechanisms. We have a tendency to describe each how to apply the projected strategy, and its effects on the target system deployed within the cloud.

Keywords: Cloud computing, stealthy attack patterns, stealthy DoS attacks.

I. 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) regulate the costs that the cloud customers have to pay for the provided quality of service (QoS). 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 (including memory, processing resources, and network bandwidth). 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. 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, or however, 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. In contrast with them, it is an iterative and incremental process. In particular, the attack potency (in terms of service requests rate and concurrent attack sources) 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 (without knowing in advance the target system capability). We show that the features offered by the cloud provider, to ensure the SLA negotiated with the customer (including the load balancing and auto-scaling 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. 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. Even if the victim detects the SIPDAS attack, the attack strategy can be re-initiate by using a different application vulnerability (polymorphism in the form), or a different timing (polymorphism over time).

II. STEALTHY DOS CHARACTERIZATION AND MODELING

This section defines the characteristics that a DDoS attack against an application server running in the cloud should have to be stealth. Regarding the 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 compared to the normal operation. Moreover, the attack is more expensive for the cloud customer and/or cloud provider, as higher the cloud resource consumption to process the malicious requests on the target system. From the point of view of the attacker, the main objective is to maximize the ratio between the amount of 'damage' caused by the attack (in terms of service degradation

and cloud resources consumed), and the cost of mounting such an attack (called 'budget').

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. Moreover, we assume that a load balancing mechanism dispatches the user service requests among the instances. The instances can be automatically scaled up or down, by monitoring some parameter suitable to assess the provided QoS (e.g., the computational load, the used memory, and the number of active users). Specifically, we model the system under attack with a comprehensive capability ζM, which represents a global amount of work the system is able to perform in order to process the service requests. Such capability is affected by several parameters, such as the number of VMs assigned to the application, the CPU performance, the memory capability, etc. Each service request consumes a certain amount wi of the capability ζM on the base of the payload of the service request. Thus, the load CN of the system at time t can be modeled by a queuing system M/M/n/n with Poisson arrivals, exponentially distributed service times, multiple servers, and n incoming requests in process (system capability). Moreover, the auto scaling feature of the cloud is modeled in a simple way: when new resources (e.g., VMs) are added to the system, the effect is an increase of the system capability ζM.

Therefore, given η legitimate type of service requests θ=(ϑ1, ..., ϑη), and denoted w as the cost in terms of cloud resources necessary to process the service request φ ∈ θ, an attack against a cloud system can be represented as in Fig. 1. Specifically, Fig. 1 shows a simple illustrative attack scenario, where the system is modeled as: (i) a queue (that conceptually represents the load balancing mechanism), in which are queued both the legitimate user request flows ϕNj and the DDoS flows ϕAj (attack sources), and (ii) a job for each service request that is currently processed on the system.

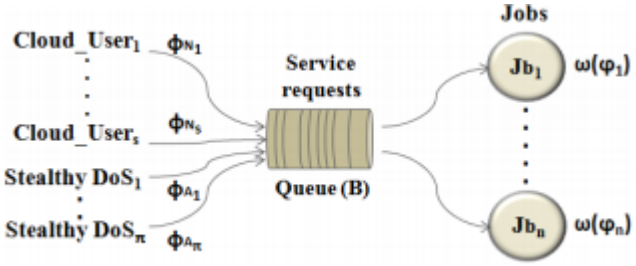

Figure 1. Attack scenario

Stealthy Attack Objectives:-

In this section, we aim at 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 (e.g., service response time), 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 (but well orchestrated and timed) 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 (the legitimate requests are continually discarded). 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 ϕAj =[φj,1,...,φj, m], that satisfy the following optimization problem:

Stealthy DDoS attack pattern in the cloud:-

Denote π the number of attack flows, and consider a time window T, the DDoS attack is successful in the cloud, if it maximizes the following functions of profit and resource consumption:

$$\begin{cases} maximize & P_A = \sum_{j=1}^{\pi} \sum_i g(\varphi_{j,i}), \\ maximize & C_A = \sum_{j=1}^{\pi} \sum_i w(\vartheta_{j,i}), \end{cases} \quad (1)$$

And it is performed in stealthy fashion, if each flow ϕAj satisfies the following conditions:

$$\begin{cases} (c_1) & minimize & \delta_j, \ \forall j \in [1..\pi], \\ (c_2) & s.t.\ to & \varphi_{j,i} \in \theta, \\ (c_3) & s.t.\ to & exhibits\ a\ pattern\ neither\ periodic\ nor\ impulsive, \\ (c_4) & s.t.\ to & exhibits\ a\ slowly\ increasing\ intensity, \end{cases} \quad (2)$$

Where:

• g is the profit of the malicious request φj, i, which expresses the service degradation (e.g., in terms of increment of average service time tS to process the user requests with respect to the normal operation);

• δj is the average message rate of the flow ϕAj ,

• W is the cost in terms of cloud resources necessary to process φj,i ∈ θ. Cond. (2.c1) implies that the flow ϕAj has to be injected with a low-rate δj .

Cond. (2.c2) assumes that all attack messages have to be legitimate service requests.

Creating Service Degradation:-

Considering a cloud system with a comprehensive capability ζM to process service requests φi , and a queue with size B that represents the bottleneck shared by the customer's flows ϕNj and the DoS flows ϕAj (Fig. 1). Denote C0 as the load at time the onset of an attack period T (assumed to occur at time t0), and CN as the load to process the user requests on the target system during the time window T. To exhaust the target resources, a number n of flows ϕAj have to be orchestrated, such that:

$$C_0(t_0) + C_N(T) + C_A(T) \geq \zeta_M * T, \quad (3)$$

Where CA (T) represents the load to process the malicious requests φi during the period T. If we assume that (1) the attack flows are not limited to a peak rate due to a network bottleneck or an attacker's access link rate, and (2) the term CN can be neglected during the attack (CA >> CN), the malicious resource consumption CA can be maximized if the following condition is verified:

$$C_A(T) \geq \zeta_M * T - C_0(t_0) \quad with\ C_A >> C_N. \quad (4)$$

Moreover, assume that during the period T, the requests φi ∈ ϕA burst at an average rate δA, whereas the flow ϕN bursts at an average rate δN. Denote B0 as the queue size at time t0, and d as the time that the queue becomes full, such that:

$$d = \frac{B - B_0}{\delta_A + \delta_N - \delta_p}, \quad (5)$$

Where δp is the average rate of requests processed on the target system (i.e., the system throughput during the period T). After d seconds, the queue remains full if δA +δN ≥ δp. In particular, under attack, if d < T and CA (Ω) ≥ ζM ∗Ω–C0(t0 +d), the attacker can archive the best profit PA during the time window Ω = [t0 + d, T] (i.e., there will be a high likelihood for the user requests to be neglected, forcing the client to perform a service request retransmission).

Minimize Attack Visibility:-

According to the previous stealthy attack definition, in order to reduce the attack visibility, Conditions (2) have to be satisfied. Therefore, through the analysis of both the target system and the legitimate service requests (e.g., the XML document structure included within the HTTP messages), a patient and intelligent attacker should be able to discover an application vulnerability (e.g., a Deeply-Nested XML vulnerability), and identify the set of legitimate service request types ϑk ⊂ θ (Cond. (2.c2)), which can be used to leverage such vulnerability. For example, for an X-DoS attack, the attacker could implement a set of XML messages with different number of nested tags nTi = 1, ..., NT . The threshold NT can be either fixed arbitrarily, or possibly, estimated during a training phase, in which the attacker injects a sequence of messages with nested XML tags growing, in order to identify a possible limitation imposed by a threshold-based XML

validation schema. A similar approach can be used to estimate the maximum message rate δT with which injecting the service requests φi . Then, the attacker has to define the minimal number π of flows ϕA characterized by malicious requests injected with:

An average message rate lower than δT, in order to evade rate-controlling- and time-window-based detection mechanisms (Cond. (2.c1)), and A polymorphic pattern (described in the next section), in order to evade low-rate detection mechanisms (Conditions (2.c3 and 2.c4)), Such that maximize the functions PA and CA (Equ. (1)).

III. CONCLUSION

In this paper, we tend to propose a technique to implement furtive attack patterns that exhibit a slowly-increasing polymorphic behavior which will evade, or however, greatly delay the techniques proposed within the literature to discover low-rate attacks. Exploiting a vulnerability of the target application, a patient and intelligent wrongdoer will orchestrate refined flows of messages, indistinguishable from legitimate service requests. In specific, the planned attack pattern, rather than aiming at making the service out of stock, it aims at exploiting the cloud flexibility, forcing the services to rescale and consume a lot of resources than required, touching the cloud client a lot of on financial aspects than on the service handiness.

IV. REFERENCES

[1]. M. C. Mont, K. McCorry, N. Papanikolaou, 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]. F. Cheng, C. Meinel,"Intrusion Detection in the Cloud," in Proc. IEEE Int. Conf. Dependable, Autonom. Secure Comput. Dec. 2009, pp. 729–734.

[3]. C. Metz. (2009, Oct.),"DDoS attack rains down on Amazon Cloud OnlineAvailable: http://www.theregister. co.uk/2009/10/05/amazon_bitbucket_outage/S

[4]. K. Lu, D. Wu, J. Fan, S. Todorovic, A. Nucci,"Robust and efficient detection of DDoS attacks for large-scale internet," Comput. Netw. Vol. 51, No. 18, pp. 5036–5056, 2007.

[5]. H. Sun, J. C. S. Lui, D. K. Yau,"Defending against low-rate TCP attacks: Dynamic detection and protection," In Proc. 12th IEEE Int. Conf. Netw. Protocol, 2004, pp. 196-205.

[6]. A. Kuzmanovic, E. W. Knightly,"Low-rate TCP-Targeted denial of service attacks: The shrew Vs. the mice and elephants," In Proc. Int. Conf. Appl., Technol., Archit., Protocols Comput. Commun. 2003, pp. 75–86.

[7]. M. Guirguis, A. Bestavros, I. Matta, Y. Zhang,"Reduction of quality (RoQ) attacks on internet end-systems," In Proc. IEEE Int. Conf. Comput. Commun. Mar. 2005, pp. 1362– 1372.

[8]. X. Xu, X. Guo, 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, X. Li,"Thwarting zeroday 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, 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, R. Aversa,"Architecturing a sky computing platform," In Proc. Int. Conf. Towards Serv.-Based Int., 2011, Vol. 6569, pp. 1-13.

[12]. U. Ben-Porat, A. Bremler-Barr, H. Levy,"Evaluating the vulnerability of network

mechanisms to sophisticated DDoS attacks," In Proc. IEEE Int. Conf. Comput. Commun., 2008, pp. 2297–2305.

[13]. S. Antonatos, M. Locasto, S. Sidiroglou, A. D. Keromytis, E. Markatos,"Defending against next generation through network/ endpoint collaboration and interaction," In Proc. IEEE 3rd Eur. Int. Conf. Comput. Netw. Defense, 2008, Vol. 30, pp. 131–141.

[14]. R. Smith, C. Estan, S. Jha,"Backtracking algorithmic complexity attacks against a NIDS," In Proc. Annu. Comput. Security Appl. Conf., Dec. 2006, pp. 89–98.

[15]. C. Castelluccia, E. Mykletun, G. Tsudik,"Improving secure server performance by re-balancing SSL/TLS handshakes," in Proc. ACM Symp. Inf., Apr. 2005, pp. 26–34.

[16]. 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. of the 2th Int. Conf. on Cloud Computingand Services Science, 2012, pp. 670-674.

[17]. F. Cheng and C. Meinel. Intrusion Detection in the Cloud. In Proc. Of the IEEE Int. Conf. on Dependable, Autonomic and Secure Computing, Dec. 2009, pp. 729-734.

[18]. C. Metz. DDoS attack rains down on Amazon Cloud. Available at: http://www.theregister.co.uk/2009/10/05/amazo n bitbucket outage/S, 26 Oct. 2009.

[19]. K. Lu, D. Wu, J. Fan, S. Todorovic, and A. Nucci. Robust and efficient detection of DDoS attacks for large-scale internet. In Computer Networks, vol. 51, no. 18, 2007, pp. 5036-5056.

[20]. H. Sun, John C. S. Lui, and D. K. Yau. Defending against low-rate tcp attacks: Dynamic detection and protection. In Proc. of the 12th IEEE Int.Conf. On Network Protocols, 2004, pp. 196-205.