Cloud Computing Strategy for Overflow of Denied Data

B.VENKATESWARA REDDY
PG Student (Software Engineering)
St. Johns College of Engineering and Technology
Kurnool, Andhra Pradesh, India

U U VEERENDRA
Assistant professor, Department of CSE
St. Johns College of Engineering and Technology
Kurnool, Andhra Pradesh, India

Abstract— **The success of the cloud computing paradigm is due to its on-demand, self-service, and pay-by-use nature. According to this paradigm, the effects of Denial of Service (DoS) attacks involve not only the quality of the delivered service, but also the service maintenance costs in terms of resource consumption. Specifically, the longer the detection delay is, the higher the costs to be incurred. Therefore, a particular attention has to be paid for stealthy DoS attacks. They aim at minimizing their visibility, and at the same time, they can be as harmful as the brute-force attacks. They are sophisticated 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 propose a strategy to orchestrate stealthy attack patterns, which exhibit a slowly-increasing-intensity trend designed to inflict the maximum financial cost to the cloud customer, while respecting the job size and the service arrival rate imposed by the detection mechanisms. We describe both how to apply the proposed strategy, and its effects on the target system deployed in the cloud.**

Index Terms—**Cloud Computing, Sophisticated Attacks Strategy, Low-Rate Attacks, Intrusion Detection**

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) [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 (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 a 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 BitBucket 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.

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 [5], [6], [7], [8]. 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 [9]. 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).

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 (to cause a significant financial losses), and evades, or however, delays greatly the detection methods. Moreover, 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 XML-based 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].

The rest of this paper is organized as follows. Background and related work are presented in Section 2. Section 3 illustrates several examples of attacks, which can be leveraged to implement the proposed attack pattern. Section 4 describes the proposed strategy to build the stealthy attacks, and presents the attack pattern, whose detailed implementation is reported in Section 5. Section 6 introduces the X-DoS attack used as case study. Section 7 shows the experimental results obtained running the attack pattern. Validation of stealthy characteristics is provided in Section 8. Some considerations about countermeasures against the proposed strategy are illustrated in Section 9. Conclusions and future work are described in Section 10.

II. BACKGROUND AND RELATED WORK

2.1 Related Work

Sophisticated DDoS attacks 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 [7], [12]. 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. These attacks can be significantly harder to detect compared with more traditional brute-force and flooding style attacks [5].

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 Oð1Þ [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 a low-rate traffic in order to be unnoticed by the DDoS detection mechanisms. In recent years, variants of DoS attacks that use low-rate traffic have been proposed, including Shrew attacks (LDoS), Reduction of Quality attacks (RoQ), and Low-Rate DoS attacks against application servers (LoRDAS).

The terminology 'stealthy DDoS' mainly refers to Shrew attacks first introduced in [6], which was followed by a series of related research [18], [19]. It refers to periodic, pulsing, and low-rate attack traffic against the TCP protocol. Specifically, it has been used to exploit TCP's retransmission time-out (RTO) mechanism, provoking a TCP flow to repeatedly enter in a RTO state. This is obtained by sending high rate but short-duration bursts (having round trip time scale burst length), and repeating periodically at slower RTO time-scales [5], [16]. RoQ attacks target the dynamic operation of the adaptation mechanisms widely adopted to ensure that the workload would be distributed across the system resources to optimize the overall performance [7], [12], [20]. By using a specific attack pattern, RoQ induces constant oscillations between the overload and underload states, without sustaining the attack traffic. This is achieved by timing the attack traffic and its magnitude in order to exploit the dynamics of the system.

As for LoRDAS, the applications are usually more vulnerable to DDoS attacks due to their complexity. Specifically, LoRDAS attacks have no necessary deviations in terms of network traffic volumes or traffic distribution with respect to normal traffic. Due to its high similarity to legitimate network traffic and much lower launching overhead than classic DDoS attack, 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. Several LoRDAS attack models against application server have been proposed [8], [23], [24]. In particular, they aim at keeping the service queue of the target application servers completely full of requests coming from the attacker, so that any new incoming request sent by legitimate users is discarded. Macia-Fernandez et al. [24] present an evolution of the low-rate DDoS attack against iterative application servers, and extends its capabilities to concurrent systems. They assume the target server has a finite service queue, where the incoming service requests are temporarily stored to be served by the corresponding application process or thread. 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. The actual availability of the service is thus reduced, while the data rate is low to elude potential defense mechanisms deployed against high-rate DDoS attacks at the server side. However, the major attention was paid to the mechanism for forecasting of the application server response times. Regarding this, Xiaodong et al. [8] propose a modified strategy, according to which the attack differs in behavior at different stage of the process. Specifically, the attack process has two different stages, before and after the service queue is filled up. The attacker monitors the number of attack requests that have been declined by the server within a recent period. If the requests are widely rejected, the attacker can assume that the service queue has been filled, and the attack is suspended. However, both the described attacks exhibit the typical periodic waveform of the low-rate DDoS attack, which consists of ON-OFF sequences of attack messages. Several works have proposed approaches to detect attacks that exhibit such a periodic and pulsing behavior [25], [26], [27].

2.2 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 customers can pay 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. For example, by using Amazon EC2 cloud services, the consumers can set a condition to add new computational instances when the average CPU utilization exceeds a fixed threshold.

Moreover, they can configure a cool-down period in order to allow the application workload to stabilize before the auto scaling adds or removes the instances [29]. In the following, we will show how this feature can be maliciously exploited by a stealthy attack, which may slowly exhaust the resources provided by the cloud provider for ensuring the SLA, and enhance the costs incurred by the cloud customer.

III. 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[10], [12], [30], 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 WS-Security specification (e.g., an oversized security header of a SOAP message can cause the same effects of an Oversize Payload, as well as a chained encrypted key can lend to high memory and CPU consumptions); 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 (i.e., messages that can be discarded by the system, but at the expense of a huge amount of redundant work, such as the Business Process Execution Language (BPEL) based document, which must be read and processed completely, before they may safely be discarded); 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.

IV. 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 the cost of mounting such an attack (called 'budget') [7].

4.1 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 zM, 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 zM 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 zM.

Therefore, given η legitimate type of service requests θ =(ϑ1,...,ϑh), and denoted w as the cost in terms of cloud resources necessary to process the service request $\varphi \in \theta$, 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 fNj and the DDoS flows fAj (attack sources), and (ii) a job for each service request that is currently processed on the system.

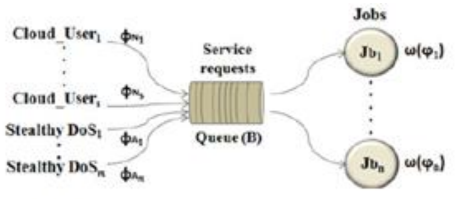

Fig. 1. Attack scenario.

4.2 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 [18], [25], [26]. 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.

V. 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.

In the future work, we aim at extending the approach to a larger set of application level vulnerabilities, as well as defining a sophisticated method able to detect SIPDAS based attacks in the cloud computing environment.

VI. 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] F. Cheng and 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 [Online]. Available: http://www.theregister.co.uk/2009/10/05/amazon_bitbucket_outage/S

[4] K. Lu, D. Wu, J. Fan, S. Todorovic, and 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, and 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 and 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, and 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, 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.

[12] U. Ben-Porat, A. Bremler-Barr, and 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, and 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, and 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, and G. Tsudik, "Improving secure server performance by re-balancing SSL/TLS handshakes," in Proc. ACM Symp. Inf., Apr. 2005, pp. 26–34.

AUTHOR's PROFILE

B. Venkateswara Reddy received B.tech degree

from JNTU anantapur .He is currently pursuing M.tech Degree in Software Engineering specialization at St. Johns College of Engineering & Technology, Yerrakota, Kurnool, India.

U.U.VEERENDRA Received B.tech degree from

JNTU anantapur and received M.tech degree from JNTU anantapur. He is currently working as assistant professor, Department of CSE, in St. Johns College of Engineering & Technology, Yerrakota, Kurnool, Andhra Pradesh, India.