International Journal of Advanced Technology and Innovative Research

www.ijatir.org

ISSN 2348–2370
Vol.08,Issue.09,
August-2016,
Pages:1741-1744

Stealthy Denial of Service Strategy in Cloud Computing

S. SHIRISHA
Associate Professor, Dept of CSE, Sagar Institute of Technology, Chevella, TS, India, E-mail: sirisha2805@gmail.com.

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.

Keywords: Cloud Computing, Sophisticated Attacks Strategy, Low-Rate Attacks, Intrusion Detection.

I. INTRODUCTION

Cloud providers offer services to rent computation and storage capacity, in a way as transparent as possible, giving the impression of 'unlimited resource availability'. 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. In order to minimize the customer costs, the auto scaling ensures that the number of the application instances increases seamlessly during the demand spikes 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. 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.

II. EXISTING AND PROPOSED SYSTEMS

A. Existing System

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. The term stealthy has been used 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. The methods of launching sophisticated attacks can be categorized into two classes: job-content-based and jobs arrival pattern-based. 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).

B. Proposed System

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 kinds of attacks that leverage known application vulnerabilities, in order to degrade the service provided by the target application server running in the cloud.

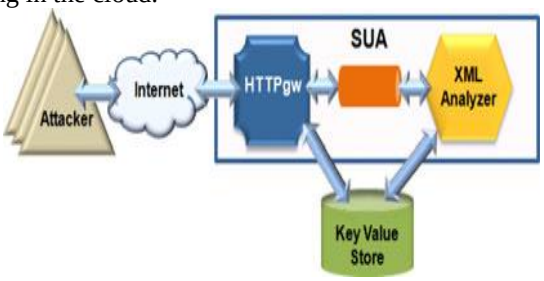

Fig.1. System Architecture.

Advantages of Proposed System:

- 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.
- Ii Even if the victim detects the attack, the attack process can be re-initiate by exploiting a different application vulnerability (polymorphism in the form), or a different timing (polymorphism over time), in order to inflict a prolonged consumption of resources.

III. 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). 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 Bit Bucket Cloud, which went down for 19h. Therefore, the cloud management system has to implement specific counter measures 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.

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 kinds 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 reinitiated by using 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 overloads 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 XML-based DoS (XDoS) attacks to the web-based systems. The experimental test bed 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.

IV. ATTACK EVALUATION

In this section, first we present an example of attack implemented by using the paradigm offered by the mOSAIC framework. Then, we study the impact of the SIPDAS-based attack pattern on the quality of service provided by the target application, by varying the number of the involved Agents and the resources of the application server.

A. Attack Implementation

The implementation of a SIPDAS-based attack can be done in several ways. In this work, we use the same cloud framework adopted for building up the target server application SUA. As a result, the implemented attack can be offered as services through a simple web interface. Fig.1 shows the architecture of the attacker application in terms of the mOSAIC Cloudlets. A web interface is used to setup the attack parameters and observe the status of the attack. When the attack is activated by the web interface, a set of parameters is sent to the Master, including the target system URL, the attack intensity I0, the attack increment DI, the thresholds NT and dT (e.g., the maximum number of nested tags and service request rate), and the attack period T. The Master coordinates the attack, by enabling the Agent instances, and interacting with the Meter that performs legitimate requests to the server under attack, and differently from the Agents, evaluates the response time tS. The KV store shared among the Cloudlets, maintains all the information related to the attack state, including the parameters used by the Agents and the attack results (in terms of reached service degradation) evaluated by the Meter. The Master periodically acquires information from the 'KV store', and sends messages to Agents in order to update their actions, according to the attack strategy described.

B. Experimental Evaluation

In the following experiments, we assume that during the normal operation the target application SUA runs on a certain number of VMs (with 2 CPU x86, 32 bit, 2.0 Ghz with 1 GB of memory) in a mOSAIC-based private cloud. The auto-scaling mechanism is enabled by the mOSAIC Platform when the average CPU load on the involved VMs exceeds the 90 percent for a time period greater than 10 minutes. Moreover, we adopt the developed TPC-W emulator both to simulate the customer workload and to evaluate the attack effect. The TCP-W emulator and the attacker application are deployed on different VMs and connected to the target cloud through a private network (100 Mb/s Ethernet LAN). The settings for the attack evaluation are selected as following: $I_0 = 10$ (initial attack intensity), $\Delta_I = 10$ (attack intensity increment), and the maximum number of nested tags $N_T = 40$ (we assume that it is imposed by a prevention mechanism based on a validation schema). Moreover, in order to achieve a small evaluation time, the attack period is chosen to be T = 120 s. Finally, during the first two experiments the mOSAIC auto scaling mechanism is disabled.

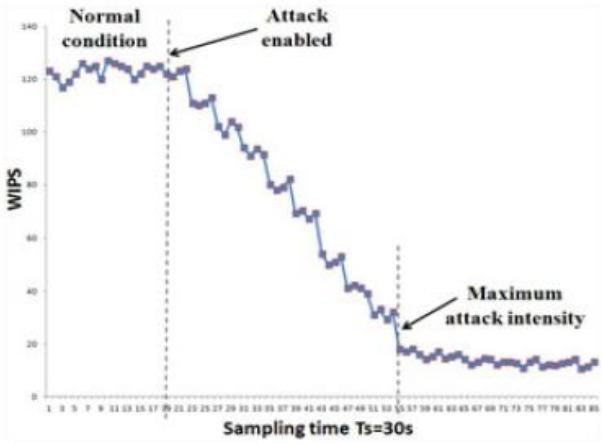

Fig.2. SIPDAS effect (with a single Agent) on the mOSAIC-based application running on a single VM (auto-scaling disabled).

During the first experiment, we evaluate the maximum message rate δ_T necessary to inflict a substantial service degradation. According to the filtering function γ described (with D = 0.98 and λ= 3), we assume that the attack is successful if $t_s(\varphi_i) > \mu_R + 3\sigma_R$ for a number of consecutive service requests greater that H = 60. In order to show the attack effects, Fig.2 shows the WIPS variation with respect to the time, achieved with a single Agent against SUA deployed on a single VM on the server side. In order to make clearer the achieved results, the WIPS values are aggregated at a fixed time interval T_S = 30 s and the average value is shown. Experimental results show that are sufficient about nine attack periods (i.e., about t = 9* T = 18 minutes) to satisfy, as well as to achieve a service degradation greater than 90 percent. The smallest reached inter-arrival time between two consecutive message (in the attack sequence) is t_I = 26ms, whereas the average value is t_I = 73ms. In the second experiment, we set the threshold δ_T to the average value t_I reached during the previous experiment (δ_T =1/73ms). Fig. 3 shows the WIPS variation achieved when the SUA is deployed on two VMs.

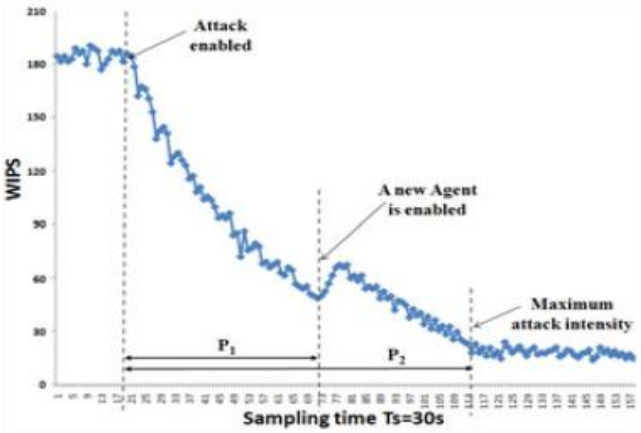

Fig.3. SIPDAS effect (with two Agents) on the mOSAIC-based application running on two VMs (auto-scaling disabled).

Results show that a single Agent is not able to inflict a significant service degradation. Specifically, at sample #73 the Agent reaches the maximum achievable attack intensity CRM with the fixed δ_T (after a period $P_1 = 50 * T_S$ from the attack activation). At this point, the Master enables another Agent and sets a new initial attack intensity of the two Agents to $I_0 = C_{RM}/=2$. As Fig3 shows, with two Agents the maximum service degradation is achieved after a time period $P_2 = 96*T_S= 48$ minutes. In the third experiment, the mOSAIC auto-scaling mechanism is enabled. We assume that in normal conditions the target application runs on two VMs, whereas in case of overloading due to a workload peak, the auto-scaling mechanism can incrementally add other five VMs. Experimental results show that after about 3 hours the attack inflicts the maximum service degradation with five Agents.

V. CONCLUSION

Cloud resources are shared with mutual and commercial models. Slowly-Increasing- Polymorphic DDoS Attack Strategy (SIPDAS) is adapted to initiate DDoS attacks on the clouds. Cloud Intrusion Detection System (CIDS) is constructed to discover the SIPDAS attacks with flow correlation analysis. Polymorphic behavior identification and cost analysis methods are integrated with the CIDS. Cloud Intrusion Detection System (CIDS) is build to discover slowly-Increasing- Polymorphic DDoS Attack Strategy (SIPDAS). The CIDS controls the resource consumption and cost factors. The system minimizes the application level vulnerabilities. Attack behavioral changes are automatically detected by the system.

VI. REFERENCES

[1] Massimo Ficco and Massimiliano Rak, "Stealthy Denial of Service Strategy in Cloud Computing", IEEE Transactions on Cloud Computing, Vol. 3, No. 1, January-March 2015.

[2] 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.

[3] F. Cheng and C. Meinel, "Intrusion Detection in the Cloud," in Proc. IEEE Int. Conf. Dependable, Autonom. Secure Comput., Dec. 2009, pp. 729–734.

[4] C. Metz. (2009, Oct.). DDoS attack rains down on Amazon Cloud [Online]. Available: http://www.the register.co.uk/2009/10/05/amazon_bitbucket_outage/S.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

Author Profile:

Mrs.S.Shirisha, received the Master of Technology degree in **Computer Science** from the Vidya Vikas Institute of Technology-JNTUH, received the Bachelor of Technology degree from Anwarul-Uloom College of Engineering & Technology-JNTUH. She is currently working as Associate Professor and a Head of the Department of CSE with Sagar Institute of Technology, Chevella and previously worked as Senior Asst.Prof of CSE with Anwarul-Uloom College of Engineering & Technology. Her interest subjects are Design patterns, Operating Systems, DBMS, Design and Analysis of Algorithms and Cloud Computing etc.

Email: sirisha2805@gmail.com.