International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 01 | Jan -2017 www.irjet.net p-ISSN: 2395-0072

A Survey on Stealthy Denial of Service Strategy in Cloud Computing

MISS. DIPALEE BALU RAUT[1], PROF. MR. SAYYAD G. G[2].

[1] *PG Scholar, Departement of Computer Engineering Dattakala Faculty of Engineering, Pune Maharashtra, India*

[2] *Proffessor, Departement of Computer Engineering Dattakala Faculty of Engineering, Pune Maharashtra, 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.***

***Key Words*: Cloud computing, sophisticated attacks strategy, low-rate attacks, intrusion detection.**

1.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.

2. LITERATURE SURVEY

We present in this paper the novel concept of a policy orchestration service, which is designed to facilitate security and privacy governance in the enterprise, particularly for the case where various services are provided to the enterprise through external suppliers in the cloud. The orchestration service mediates between the enterprises' internal decision

support systems (which incorporate core security and privacy recommendations) and the cloud-based service providers, who are assumed to be bound by contractual service level agreements with the enterprise. The function of the orchestration service, which is intended to be accessed as a trusted service in the cloud, is to ensure that applicable security and privacy recommendations are auctioned by service providers through adequate monitoring and enforcement mechanisms[1].

Cloud computing is a new business model, which represents an opportunity for users, companies, and public organisations to reduce costs and increase efficiency, as well as an alternative way for providing services and resources. In this pay-by-use model, security plays a key role. Cyber attacks are a serious danger, which can compromise the quality of the service delivered to the customers, as well as the costs of the provided cloud resources and services. In this paper, a hybrid and hierarchical event correlation approach for intrusion detection in cloud computing is presented. It consists of detecting intrusion symptoms by collecting diverse information at several cloud architectural levels, using distributed security probes, as well as performing complex event analysis based on a complex event processing engine. The escalation process from intrusion symptoms to the identified cause and target of the intrusion is driven by a knowledge-base represented by an ontology. A prototype implementation of the proposed intrusion detection solution is also presented[2].

3. 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 behaviours 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).

Disadvantages:

1. 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.
2. The methods of launching sophisticated attacks can be categorized into two classes: job-content-based and jobs arrival pattern-based.
3. 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).

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

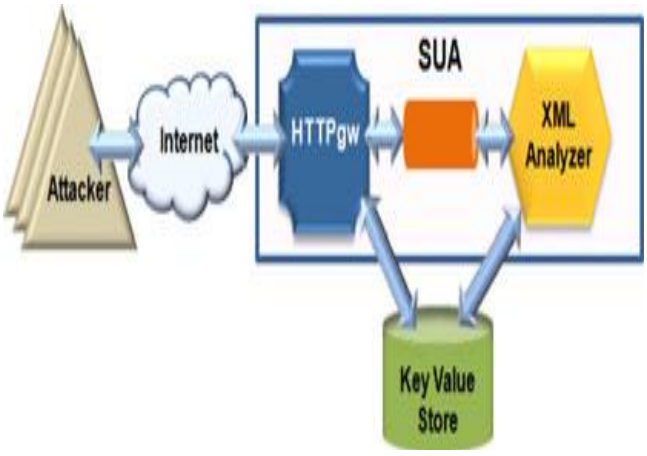

Fig -1: Block Diagram of Proposed System

5. MODELS

A. Server under Attack Model
B. Creating Service Degradation
C. Minimize Attack Visibility
D. XML-Based DoS Attack

MODULES DESCSRIPTION

A. 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, 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 of the capability 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 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.

B. Creating Service Degradation:

Considering a cloud system with a comprehensive capability to process service requests, and a queue with size B that represents the bottleneck shared by the customer's flows

and the DoS flows. 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 have to be orchestrated.

C. Minimize Attack Visibility:

According to the previous stealthy attack definition, in order to reduce the attack visibility, Conditions 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, 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. 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 with which injecting the service requests.

D. XML-Based DoS Attack

During the experimental campaign, we analyzed the CPU consumption depending on the number of nested XML tags and the frequency with which the malicious messages are injected. In particular, the CPU consumption on the target system to parse messages containing XML tags with different nested depth. [t] The results showed that a message of 500 nested tags is sufficient to produce a peak of CPU load of about 97 percent, whereas with 1,000 tags the CPU is fully committed to process the message for about 3 seconds. Moreover, we performed several attacks. For each attack, we injected a homogeneous XDoS flow, i.e., a sequence of messages with a fixed number of nested tags and a fixed message rate. Assuming that 20 seconds are the maximum time observed experimentally to reach a steady state value of CPU under attack (namely CR), and denoting 'baseline' as the average CPU load in absence of user load (about 9 percent). 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.

ADVANTAGES

- 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.

- 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.

6. CONCLUSION

In this paper, we propose a strategy to implement stealthy attack patterns, which exhibit a slowly increasing polymorphic behaviour 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, in distinguishable 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.

ACKNOWLEDGEMENT

I would like to express my gratitude towards my guide and H.O.D of Computer Engineering Department Prof. Sayyad G.G. for their support and guidance in my work.

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] M. Ficco, "Security event correlation approach for cloud computing," Int. J. High Perform. Comput. Netw., vol. 7, no. 3, pp. 173–185, 2013.

[3] 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.theregister.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] 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.