ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Strategy to Organize Stealthy Denial of Service Attack Patterns in Cloud Computing

L.Shirisha
PG.Scholar
Department of CSE
Anurag Group of Institutions
Hyderabad.

A.Obulesu
Assistant Professor
Department of CSE
Anurag Group of Institutions
Hyderabad.

Dr.G.Vishnu Murthy
Professor & HoD
Department of CSE
Anurag Group of Institutions
Hyderabad.

Abstract:
Over the past decade, many attempts have been committed to the detection of DDoS attacks in distributed systems. Security prevention mechanisms often use approaches based on rate-controlling, time-window, worst-case threshold, and pattern-matching methods to distinguish between the nominal system operation and malicious behaviors. They attempt to fulfill their activities in a "stealthy" fashion in order to circumvent the security mechanisms, by orchestrating and timing attack patterns that hold definite weaknesses of target systems and the amount of time that the ongoing attack to the system has been undetected. Here enlightened strategy is presented to orchestrate stealthy attack patterns against applications running in the cloud platform. In preference to aiming at making the service unavailable, the proposed strategy focuses at exploiting the cloud flexibility, forcing the application to use more resources than needed, disturbing the cloud customer more on financial facets than on the service availability.

Index Terms: ***Cloud computing, sophisticated attacks strategy, intrusion detection***

1.Introduction

Cloud computing can serve enterprises growth the creation and delivery of IT solutions by producing them with access to services in a cost-effective and adjustable manner. Clouds can be classified into three categories, based on their convenience sectors and the deployment model. They are: Public Cloud, Private Cloud and Hybrid Cloud. A public Cloud is made available in a pay-as-you-go mode to the general public users irrespective of their original association. A private Cloud's usage is cramped to members, employees, and trusted partners of the organization. A hybrid Cloud authorizes the use of private and public Cloud in a seamless manner. Cloud computing applications span many domains, including business, technology, government sectors, health care, smart grids, intelligent transportation networks, life sciences, automation, data analytics, consumer, disaster management and social networks. Several models for the creation, deployment, and deliver of these applications as Cloud services have emerged. To compose data management scalable in cloud computing, reduplication has been a well-known technique and has fascinated more and more attention in recent times. Service level agreements (SLA) descriminalize the costs that the cloud customers have to remunerate for the provided quality of service (QoS). A side effect of such a replica is that, it is flat to Denial of Service (DoS) and Distributed DoS (DDoS), which aspire at tumbling the service availability and concert by grueling 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. Explicitly, in cloud computing also partial service humiliation due to an attack has direct effect on the service costs, and not only on the performance and availability pretended by the customer. The delay of the cloud service provider to discover the causes of the service degradation is capable of considered as security vulnerability.

2. Cloud Resources Provisioning:

Cloud providers proffer services to rent computation and storage capacity, in a way as transparent as

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 propose the 'load balancing' service for automatically distributing the incoming application service requests across many instances, as well as the 'auto scaling' service for enabling consumers to closely follow the demand curve for their applications (reducing the need to obtain cloud resources in advance). In order to minimize the customer costs, the auto scaling protects that the number of the application instances increases seamlessly during the demand spikes (to keep the contracted performance), and decreases automatically during the demand lulls. For example, by using Amazon EC2 cloud services, the consumers can place 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 connects or removes the instances. In the following, we will show how this feature can be maliciously utilized by a stealthy attack, which may slowly exhaust the resources provided by the cloud provider for ensuring the SLA, and upgrade the costs incurred by the cloud customer.

mOSAIC Framework:

The mOSAIC project focused at offering a simple way to develop and manage applications in a multi-cloud environment. It produces a framework composed of two main components: the cloud agency and the software platform. The cloud agency behaves as a provisioning system, brokering resources from a federation of cloud providers. The mOSAIC user develops the application on its local machine, then it uses a local instance of the cloud agency in order to start-up the process of remote resource acquisition and to locate the Software Platform and the developed application. The Platform enables the execution of the developed applications on the acquired cloud resources. A Java-based API is provided to develop software ingredients in the form of Cloudlets. A mOSAIC application is a collection of Cloud lets, which are interconnected through communication resources, such as shared key value stores. The Cloudlets run on a dedicated operating system, denominated mOSAIC Operating System (mOS), which is a small Linux distribution. At runtime, the Software Platform transparently scales the Cloudlets instances on the obtained virtual machines (VM) on the base of the resource consumption (auto scaling). As an example, when the Platform perceives that a Cloudlet is overloaded (e.g., it has too messages on the inter communicating queues), it may select to start a new Cloudlet instance. The Platform assumes such a decision on the base of policies defined by the application developer (through specific mOSAIC features). Finally, a load balancing mechanism automatically balances the application service appeals among the instances.

3. Stealthy attack:

The motivation of the attack against cloud applications is not to necessarily deny the service, but rather to impose noteworthy squalor in some facet of the service, namely attack profit PA, in order to maximize the cloud resource consumption CA to process malicious requests. In order to escape the attack detection, dissimilar attacks that use low-rate traffic have been existing in the literature. Therefore, several works have proposed techniques to distinguish low-rate DDoS attacks, which observe anomalies in the instability of the incoming traffic through either a timeout frequency-domain analysis. They assume that, the chief anomaly can be incurred during a low-rate attack is that, the incoming service requests fluctuate in a extreme manner during an attack. The uncharacteristic vacillation is a combined result of two different kinds of behaviors a periodic and impulse trend in the attack pattern, and the fast decline in the incoming traffic volume. Accordingly, in order to perform the attack in stealthy fashion with respect to the proposed detection techniques, an attacker has to

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

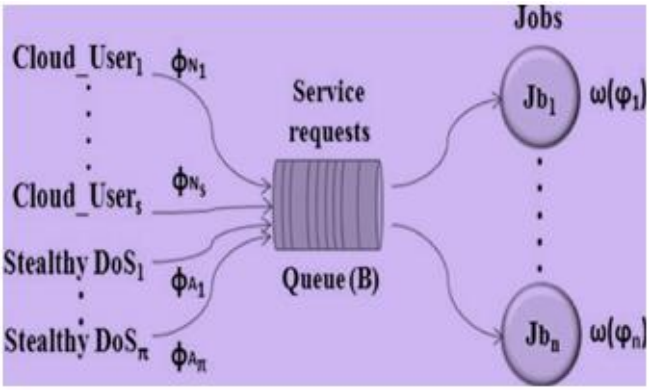
inject low-rate message flows that satisfy the optimization problem.

DOS attacks against cloud applications:

In this section are represented several attack examples, which can be leveraged to implement the proposed SIPDAS attack design against a cloud application. In particular, we consider DDoS attacks that exploit application vulnerabilities, including the Oversize Payload attack that utilizes the high memory consumption of XML processing; the bigger than the usual size cryptography that exploits the flexible usability of the security elements defined by the WS-Security specification (example: 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 supply to high memory and CPU consumptions); the Resource Exhaustion attacks use flows of messages that are correct concerning their message structure, but that are not properly correlated to any existing process instance on the quarry server (i.e., messages that can be discarded by the system, but at the expense of a huge amount of unnecessary work, such as the Business Process Execution Language (BPEL) based document, which must be read and processed completely, before they may safely discarded) and attacks that exploit the worst-case performance of the system, for example by obtaining the worst case complexity of Hash table data structure, or by using complex queries that force to spend plentiful 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 threats for the cloud applications. It exploits the XML verbosity and the complex parsing process (by using a big number of namespace declarations, oversized prefix names or namespace URIs). In particular, the Deeply-Nested XML is a resource fatigue attack, which exploits the XML message format by inserting a large number of nested XML tags in the message body. The objective is to force the XML parser within the application server, to exhaust the computational resources by processing a big number of deeply-nested XML tags.

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. Concerning the quality of service provided to the user, we assume that the system performance under a DDos attack is more demeaned, as higher the average time to process the user service requests compared to the normal operation. Additionally, the attack is more expensive for the cloud customer and/or cloud provider, as higher the cloud resource consumption to process the malicious appeals 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 ingested), and the cost of mounting such an attack (called ‘budget’).



Therefore, the first requirement to design an efficient DDos attack pattern is the ability of the attacker to assess the damage that the attack is imposing to the system, by spending a specific budget to produce the malicious additional load. the attack damage is a role of the ‘attack potency’, which depends on the number of concurrent attack sources, the request-rate of the attack flows are processed, and the job-content associated to the service requests are to be processed. Moreover, in order to make the attack stealthy, the attacker has to be able to estimate the maximum attack potency to be conducted, without that the attack pattern exhibits a behavior that may be considered anomalous by the mechanisms used as a protection for the quarry system in the following sections, starting from a synthetic representation of the quarry system,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

we narrate the conditions the attack pattern has to satisfy to minimize its visibility as long as possible, and effectively affect the target system performance in the cloud environment

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, orderly to maximize the cloud resource consumption CA to process malicious requests. In sequence 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], 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. Consequently, in order to perform the attack in stealthy fashion concerning the proposed detection techniques, an attacker has to inject low-rate message flows φAj = [φj,1, . . . , φj,m], that satisfy the following optimization problem. Attack Approach In order to implement SIPDAS-based attacks, the following components is involved:

- a Master that coordinates the attack;
- π Agents that perform the attack; and
- a Meter that evaluates the attack effects.

The proposition implemented by each Agent to perform a stealthy service humiliation in the cloud computing. It has been specialized for an X-DoS attack. Specifically, the attack is performed by introducing polymorphic bursts of length T with an increasing intensity till 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 ΔCR = ΔI as the increment of the attack intensity. For each attack period, fixed the max number of nested tags (tagThreshold), the routine pick Random Tags (. . .) randomly yields 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. 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 till either the attack is detected or the auto scaling mechanism authorized in the cloud attaches new cloud resources. The attack is performed till 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 as stated by the final level of load CR reached.

4. 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 demeaned, as higher the average time to process the user service requests 3.2 Server Under Attack Model In order to evaluate 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.

5. Conclusions:

In this paper, we suggest a strategy to implement stealthy attack patterns, exploiting a vulnerability of the target application, a patient and intelligent attacker can mobilize sophisticated flows of messages,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

indistinguishable from legitimate service requests. Specifically, the proposed attack pattern, instead of aiming at making the service unavailable, it targets at exploiting the cloud flexibility, forcing the services to scale up and consume more resources than needed, infects 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. In exacting, the proposed attack pattern, as an alternative of aiming at making the service unavailable, it aims at exploiting the cloud flexibility, efforting the services to scale up and munch through more resources than needed, distressing the cloud customer more on financial facets than on the service availability.

In the future expectations, extending the approach to a huge set of application level vulnerabilities, as well as defining a sophisticated method able to detect SIPDAS based attacks in cloud computing environment has to be focused.

References:

[1] M. Abe, M. Ohkubo, and K. Suzuki, 1-out-of-n signatures from a variety of keys, in Proc. 8th Int. Conf. Theory Appl. Cryptol. Inform. Security: Adv. Cryptol., 2002, pp. 415–432.

[2] R. Anderson, Two remarks on public-key cryptology, Manuscript, Sep. 2000. (Relevant material presented by the author in an invited lecture at the Fourth ACM Conference on Computer and Communications Security, 1997.)

[3] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, A practical and provably secure coalition-resistant group signature scheme, in Proc. 20th Annu. Int. Cryptol. Conf. Adv. Cryptol., 2000, pp. 255–270.

[4] M. H. Au, J. K. Liu, T. H. Yuen, and D. S. Wong, ID-based ring signature scheme secure in the standard model, in Proc. 1st Int. Workshop Security Adv. Inform. Comput. Security, 2006, pp. 1–16.

[5] A. K. Awasthi and S. Lal, Id-based ring signature and proxy ring signature schemes from bilinear pairings, CoRR, vol. abs/cs/ 0504097, 2005.

[6] M. Bellare, D. Micciancio, and B. Warinschi, Foundations of group signatures: Formal definitions, simplified requirements and a construction based on general assumptions, in Proc. 22nd Int. Conf. Theory Appl. Cryptographic Techn., 2003, Vol. 2656, pp. 614–629.

[7] M. Bellare and S. Miner, A forward-secure digital signature scheme, in Proc. 19th Annu. Int. Cryptol. Conf., 1999, Vol. 1666, pp. 431–448.

[8] J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, Security and privacy-enhancing multicloud architectures, IEEE Trans. Dependable Sec. Comput., 10(4) pp. 212–224, Jul. \Aug. 2013.

[9] A. Boldyreva, Efficient threshold signature, multi signature and blind signature schemes based on the gap Diffie-Hellman group signature scheme, in Proc. 6th Int. Workshop Theory Practice PublicKey Cryptography: Public Key Cryptography, 2003, Vol. 567, pp. 31–46.

[10] D. Boneh, X. Boyen, and H. Shacham, Short group signatures, in Proc. Annu.Int. Cryptol. Conf. Adv. Cryptol., 2004, Vol. 3152, pp. 41–55.

[11] E. Bresson, J. Stern, and M. Szydlo, Threshold ring signatures and applications to ad-hoc groups, in Proc. 22nd Annu. Int. Cryptol. Conf. Adv. Cryptol., 2002, Vol. 2442, pp. 465–480.

[12] J. Camenisch, Efficient and generalized group signatures, in Proc. Int. Conf. Theory Appl. Cryptographic Techn., 1997, Vol. 1233, pp. 465–479.

[13] N. Chandran, J. Groth, and A. Sahai, Ring signatures of sublinear size without random oracles, in Proc. 34th Int. Colloq. Automata, Lang. Programming, 2007, Vol. 4596, pp. 423–434.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[14] K. Chard, K. Bubendorfer, S. Caton, and O. F. Rana, Social cloud computing: Avision for socially motivated resource sharing, EEE Trans.Serv.Comput, 5(4), pp.551–563, Fourth Quarter2012.

[15] Dr. V. Goutham and M. Tejaswini, A Denial of Service Strategy To Orchestrate Stealthy Attack Patterns In Cloud Computing, International Journal of Computer Engineering and Technology, 7(3), 2016, pp. 179–186.