ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

BEHAVIORAL PATTERN BASED ATTACK DETECTION SCHEME FOR CLOUDS

Ms. C. Janani[1], Mr. S. Vijay Anand[2], Dr. T. Senthil Prakash[3]
PG Scholar – CSE[1], AP/CSE[2], Professor&HOD[3]
Shree Venkateshwara Hi-Tech Engineering College, Gobi, Tamilnadu, India
jananihicet13@gmail.com, anand.vijay87@gmail.com, jtyesp@yahoo.co.in

Abstract

Resources and services can be accessed through the cloud computing environment. 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. The Cloud Intrusion Detection System (CIDS) is build to discover Slowly-Increasing- Polymorphic DDoS Attack Strategy (SIPDAS) based attacks. Attack detection process is integrated with the service degradation and resource consumption cost analysis mechanism. Flow correlation analysis scheme is tuned to discover the polymorphic behavior of the cloud users. The attack detection model is embedded with application server to protect resource vulnerabilities.

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. Related Work

We briefly outline some masquerade detection approaches. The uniqueness approach assumes that commands that have not been seen in the training data indicate a masquerader. Moreover, the probability that a masquerader has issued a command is inversely related to the number of users that use such a command. While uniqueness has a relatively poor performance, it is one of the few approaches that target false alarm rate of 1 percent [5]. Naive Bayes One-step Markov is based upon one-step transitions from a command to the next. It builds two transition matrices for each user from, respectively, the training database and the testing one and it triggers an alarm when these matrices noticeably differ. The false alarm rate of this method is not satisfactory.

The Hybrid Multi-Step Markov method is based on Markov chains. When a Markov model cannot be adopted because too many commands in the testing data have not been observed in the training, a simple independence model with probabilities estimated from a contingency table of users versus commands may be more appropriate. Schonlau et al. toggled between a Markov model and the simple independence one. This approach achieves the best performance among the considered methods. The main idea underlying the compression approach is that new and old data from the same user should compress at about the same ratio. Instead, data from a masquerading user will compress at a different ratio. Among the proposed methods, this results in the worst performance. Incremental Probabilistic Action Modeling (IPAM) is based upon one-step command transition. It estimates the probability of each transition from the training data set and uses it to predict the sequence of user commands. Too many false predictions signal a masquerader. This method is in the lowest-performing group. Sequence-matching computes a similarity match between the user profiles and the corresponding sequence of commands. Any score lower than a threshold signals a masquerader. Its performance on the SEA data set is not very high.

Support Vector Machine (SVM) denotes a set of machine learning algorithms for binary data classification. It exploits a set of support vectors in the training data that outlines a hyper plane in feature space. SVM can potentially learn a large set of patterns but it results in high false alarm rates and a low detection rate [4]. Furthermore, the user profile has to be updated to reduce false alarms. Szymanski and Zhang propose a recursive data mining approach that discovers frequent patterns in the sequence of user commands, encodes them with unique symbols, and rewrites the sequence with the new coding. Then, a one-class SVM classifier detects masqueraders. This approach demands mixing user data and may not be ideal or easily implemented in real-world. It also suffers of some of the SVM shortcomings. Maxion and Townsend applied a Naive Bayes classifier widely used in text classification tasks and that classifies sequences of user-command data into either legitimate or masquerader. The method has not yet achieved the level of accuracy for practical deployment. Dash et al. [6] introduced an episode based Naive Bayes technique that extracts meaningful episodes from a long sequence of commands.

The Naive Bayes algorithm identifies these episodes either as masquerade or normal according to the number of commands in masquerade blocks. The proposed technique significantly improves the hit ratio but it still has high false positive rates and it does not update the user profile. Alok et al. integrates a Naive Bayes approach with one based on a weighted radial basis function, WRBF, similarity. The Naive Bayes algorithm includes information on the probabilities of commands by one user over the other users. Instead, the WRBF similarity takes into account the similarity measure based on the frequency of commands, f, and the weight associated with the frequency vectors. Here, f is a similarity score between an input frequency vector and a frequency vector from the training data set. The experiments confirm that WRBF-NB significantly improves the hit ratio but, as the previous approach, it suffers from the high false positive rates. Furthermore, it increases the overall overhead by computing both the Naive Bayes and the WRBF and integrating their results. Lastly, it does not update the user profile and neglects the low level representation of user commands. Dash et al. [6] introduced an adaptive Naive Bayes approach based on the premise that both the commands of a legitimate user and those of an attacker may differ from the trained signature but the deviation of the legitimate user is momentary, whereas the attacker one persists longer. The improvement in the performance of detection has been empirically verified using several data sets. The false positive rate is still high.

Malek and Salvatore [2] have modeled user OS commands as bag-of-words without timing information. They used a one-class support vector machine to achieve a better performance than threshold based comparison with a distance metric. The ability of sequence alignment algorithms to find areas of similarity can be exploited to differentiate legitimate usage from masquerade attacks. To do so, a signature of the normal user behavior should be created by collecting sequences of audit data. Then, this signature is aligned with audit data from monitored sessions to find areas of similarity. Areas that do not align properly are assumed to be anomalous, and several anomalous areas are a strong indicator of masquerade attacks. Among sequence alignment algorithms such as global, local and semi-global alignments, the most efficient one is semi-global alignment. Adesina et al. [3] modified the scoring function of the semi-global alignment algorithm to improve the detection efficiency. They used a systematically generated ASCII coded sequence audit data from Windows and UNIX systems as simulations for the intrusion data set. A real time evaluation using one of the current data sets is missing.

Coull et al. modified the Smith-Waterman alignment algorithm a semi-global alignment algorithm that is described with some evolutions and enhancements [7]. Different techniques in terms of the Receiver Operator Characteristic (ROC) curves and the Maxion-Townsend cost function and it shows that SGA achieves the best performances. The Maxion-Townsend cost function rates a masquerade detection algorithm and defines the detection cost.

3. Handling Denial Of Service Attacks

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. 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 BitBucket 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, 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 [8]. 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 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 reinitiate by using a 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 overload 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 [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].

4. Problem Statement

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. The following drawbacks are identified from the existing system.

- SIPDAS based attack detection is not supported
- Polymorphic behavior identification is not adapted
- Application level vulnerability detection is low
- Service degradation and resource consumption cost analysis is not performed

5. Behavioral Pattern based Attack Detection Scheme

The Cloud Intrusion Detection System (CIDS) is constructed to discover the DDoS attacks against the resource provider. Attack behaviors are dynamically changed in the Slowly-Increasing- Polymorphic DDoS Attack Strategy (SIPDAS). Cost analysis functions are integrated with the CIDS to estimate the attack levels. The system is divided into four major modules. They are Cloud Resource Provider, Service Request Analysis, Attack Detection and Cost Analysis.

Cloud resource providers are constructed to provide resources to the users. Service request flow is analyzed under the service request analysis. DDoS attacks discovered under the attack detection module. Service level and resource level cost factors are analyzed under the cost analysis.

5.1. Cloud Services

Cloud resource providers are deployed to provide the hardware and software resources. Resources are allocated with reference to the user requests. Resources are allocated with reference to the capacity and duration details. Resources are scheduled with request priority levels.

5.2. Service Request Analysis

Service requests are collected from the users. Requested user and time interval details are used in the service request analysis. Request flow is estimated using the flow correlation analysis. Request flow is compared with other user request sequences.

5. 3. Attack Detection

The attack detection process is initiated to discover the DDoS attacks. Request flow measure is verified to discover the request types. Request behavior analysis is performed to detect the polymorphic behavior of the users. Request groups are identified with similar request flows.

5.4. Cost Analysis

Cost analysis is carried out to estimate the economical impact of the attacks. Service degradation and resource consumption factors are used in the cost analysis. Resource vulnerability is also estimated in the cost analysis process. Behavioral changes are monitored with cost functions.

6. 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.

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] S. Malek and S. Salvatore, “Detecting masqueraders: A comparison of one-class bag-of-words user behavior modeling techniques,” in Proc. 2nd Int. Workshop Managing Insider Security Threats, Morioka, Iwate, Japan. Jun. 2010, pp. 3–13.

[3] A. S. Sodiya, O. Folorunso, S. A. Onashoga, and P. O. Ogundeyi, “An improved semi-global alignment algorithm for masquerade detection,” Int. J. Netw. Security, vo1. 12, no. 3, pp. 211–220, May 2011.

[4] Yongdong Wu, Zhigang Zhao, Feng Bao and Robert H. Deng, “Software Puzzle: A Countermeasure to Resource-Inflated Denial-of-Service Attacks”, IEEE Transactions On Information Forensics And Security, Vol. 10, No. 1, January 2015

[5] Hisham A. Kholidy, Fabrizio Baiardi and Salim Hariri, “DDSGA: A Data-Driven Semi-Global Alignment Approach for Detecting Masquerade Attacks”, IEEE Transactions On Dependable And Secure Computing, Vol. 12, No. 2, March/April 2015

[6] Subrat Kumar Dash, K. S. Reddy, and K. A. Pujari, “Adaptive Naive Bayes method for masquerade detection”, Security Commun. Netw., vol. 4, no. 4, pp. 410–417, 2011.

[7] Guojun Wang, Felix Musau, Song Guo and Muhammad Bashir Abdullahi, "Neighbor Similarity Trust against Sybil Attack in P2P E-Commerce", IEEE Transactions On Parallel And Distributed Systems, Vol. 26, No. 3, March 2015

[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.