ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 5, May 2016

Orchestration and detection of stealthy DoS/DDoS Attacks

Mohammedshahzan A Mulla[1], Asst prof Shivraj V B[2]

Mtech - Dept. of CSE CMRIT Bangalore.

***Abstract*— The accomplishment of the cloud computing model is due to its on-demand, self-service, & pay-by-use nature. Agreeing to this model, the effects of Denial of Service (DoS) attacks consist of not only the feature of the delivered service, but also the service care costs in terms of resource intake. Moreover, the longer the detection delay is, the more the costs to be sustained. So, a particular consideration has to be paid for stealthy Denial of Service attacks. In this project, we propose a strategy to orchestrate stealthy attack patterns, which show a slowly-increasing-intensity trend intended to impose the maximum financial cost to the cloud customer, while concerning the job size & the service arrival rate forced by the detection mechanisms. We refer to both how to apply the proposed attack strategy, & its effects on the target system deployed in the cloud.**

***Index Terms*— DOS attack, Stealthy DOS, Detection of DOS attack**

I. INTRODUCTION

Cloud computing, which being a developing model that allows customers to obtain cloud resources & services agreeing to an on-demand, self-service, & pay-by-use business model. The costs that the cloud customers have to pay for the provided quality of service (QOS) is regulated by the Service level agreements. The side effect of cloud computing model is that, it is very much susceptible to Denial of Service (DOS) & Distributed Denial of Service (DDOS), which aims at decreasing the service availability & performance by exhausting the resources of the service's host system (including memory, processing resources, & network bandwidth). These 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, & not only on the performance & availability experienced by the customer.

There can be delay by the cloud service provider to diagnose the causes of the service degradation (if it is due to either an attack or an overload which is very difficult to predict) can be well thought-out as a security vulnerability, that can be misused by Hackers that aim at exhausting the cloud resources (allocated to satisfy the negotiated QOS), & seriously degrading the QOS, as happened to the cloud vendor 'Bit Bucket'[1] Cloud, which went down for 19hours. Hence, in order to avoid paying credits in case of accidental or deliberate intrusion that cause violations of QOS guarantees the cloud management system has to implement specific countermeasures.

II. OBJECTIVE OF THE PROJECT

The main objective of this project is:

- The Main objective presents a sophisticated strategy to orchestrate/generate stealthy attack patterns contrary to applications running in the cloud.
- Instead of aiming at making the service unobtainable, the planned strategy aims at exploiting the cloud flexibility, forcing the application to consume more resources than needed, distressing the cloud customer more on financial aspects than on the service availability.
- The proposed attack strategy, viz slowly-Increasing-Polymorphic DDOS Attack Strategy (SIPDAS) can be applied to several kind of attacks that influence known application vulnerabilities, in order to

Manuscript received May, 2016.
***Mohammedshahzan A Mulla**, CSE, CMR institute of Technology., Bangalore, INDIA, 9538588786*
***Shivraj V B**, Asst Professor CSE, CMR institute of Technology, Bangalore, INDIA, 9964433911*

damage the service provided by the target application server running in the cloud.

III. Existing System

- Sophisticated DDOS attacks are definite as that category of attacks, which are custom-made to hurt a specific weak point in the target system strategy, in order to perform the denial of service or just too significantly damage the performance. The word stealthy has been used to identify sophisticated attacks that are specifically designed to keep the malicious behaviors practically invisible to the detection mechanisms [2]. These attacks can be meaningfully harder to detect related with more traditional brute-force & flooding style attacks.

- The methods of launching sophisticated attacks can be categorized into 2 classes: job-content-based & jobs arrival pattern-based.

IV. Limitations of the Existing system

- Due to its high similarity to genuine network traffic & much lower launching overhead than classic DDOS attack, this new attack type cannot be efficiently detected or prevented by current network-based solutions.

- They assume that the target server has a limited service queue, where the incoming service requests are for the time being stored to be served by the corresponding application process or thread. The proposed attack takes benefit of the capacity to predict the time at which the ACK msgs to incoming requests for agreed service occur. This capability is used to schedule an intelligent design in such a way that the attacked server becomes busy the most time in processing of the malicious requests instead of those from genuine users.

V. Problem statement

- The accomplishment of the cloud computing model is due to its on-demand, self-service, and pay-by-use nature.
- Moreover, the longer the detection delay is, the more the costs to be sustained. So, a particular consideration has to be paid for stealthy Denial of Service attacks. They target at reducing the visibility, and at the same period, they can be as dangerous as the brute-force attacks.
- In this project, we propose a strategy to orchestrate stealthy attack patterns[3], which show a slowly-increasing-intensity trend intended to impose the maximum financial cost to the cloud customer, while concerning the job size and the service arrival rate forced by the detection mechanisms.
- We refer to both how to apply the proposed attack strategy, and its effects on the target system deployed in the cloud.

VI. Proposed System

- This project paper presents a sophisticated approach to orchestrate/generate stealthy attack patterns against applications running in the cloud. As a substitute of aiming at making the service unavailable, the proposed strategy aims at take advantage of the cloud flexibility, compelling the application to ingest more resources than needed, affecting the cloud customer more on financial parts than on the service availability.
- The attack pattern is orchestrated/generated in order to evade, or on the other hand, greatly delay the techniques planned in the literature to detect low-rate attacks[4]. It does not show a periodic waveform characteristic of low-rate exhausting attacks.
- Using a cut down model empirically intended, we derive an expression for gradually increasing the potency of the attack, as a function of the stretched service degradation (without knowing in advance the target system capability).
- We show that the features presented by the cloud provider, to guarantee the SLA negotiated with the customer (including the load balancing &

auto-scaling mechanisms), can be exploited by the proposed stealthy attack, which slowly exhausts the resources provided by the cloud vendor, & increases the costs incurred by the customer.

- The planned attack approach, namely Slowly-Increasing- Polymorphic DDOS Attack Strategy (SIPDAS)[5] can be useful to several kind of attacks that force identified application vulnerabilities.
- The term polymorphic is enthused to polymorphic attacks which change message sequence at every successive infection in order to elude signature detection mechanisms. Even if the victim detects the SIPDAS attack, the attack approach can be re-initiate by using a different application vulnerability (polymorphism in the form), or a different timing (polymorphism over time).

VII. Advantages of Proposed System

- We show that the proposed attack algorithms slowly-increasing polymorphic behavior brings enough overload on the target system (which causes the client a significant financial losses[6]), & eludes, or however, delays greatly the detection methods.
- Even if the victim/cloud vendor detects the attack, the attack process can be re-initiate by exploiting a different application vulnerability (which is polymorphism in the form), or a different timing (which is polymorphism over time), in order to cause a prolonged consumption of resources.

VIII. Attack Approach

In order to implement SIPDAS-based attacks, the following components are involved:

- ❖ a Master that coordinates the attack;
- ❖ p Agents that perform the attack (each Agent injects a single flow of messages);
- ❖ a Meter that evaluates the attack effects.

A. SIPDAS ATTACK ALGORITHM:-

This algorithm describes the approach implemented by each Agent to perform a stealthy service degradation[7] in the cloud computing. Specifically, the attack is performed by injecting polymorphic bursts of length T with an increasing intensity until the attack is either successful or detected.

B. SIPDAS Core Algorithm

Attack Parameters

- ❖ Integer CR ← I_0 {Initial attack intensity.}
- ❖ Intensity (Interval between each Submit) ← DI {Attack intensity increment.}
- ❖ Threshold ← N_T (DOS attack threshold) //Its calculated by the average of 1st few ping attacks
- ❖ Attack increment ← I
- ❖ Total Time (Total time the attack goes on) ← T {Burst period.}

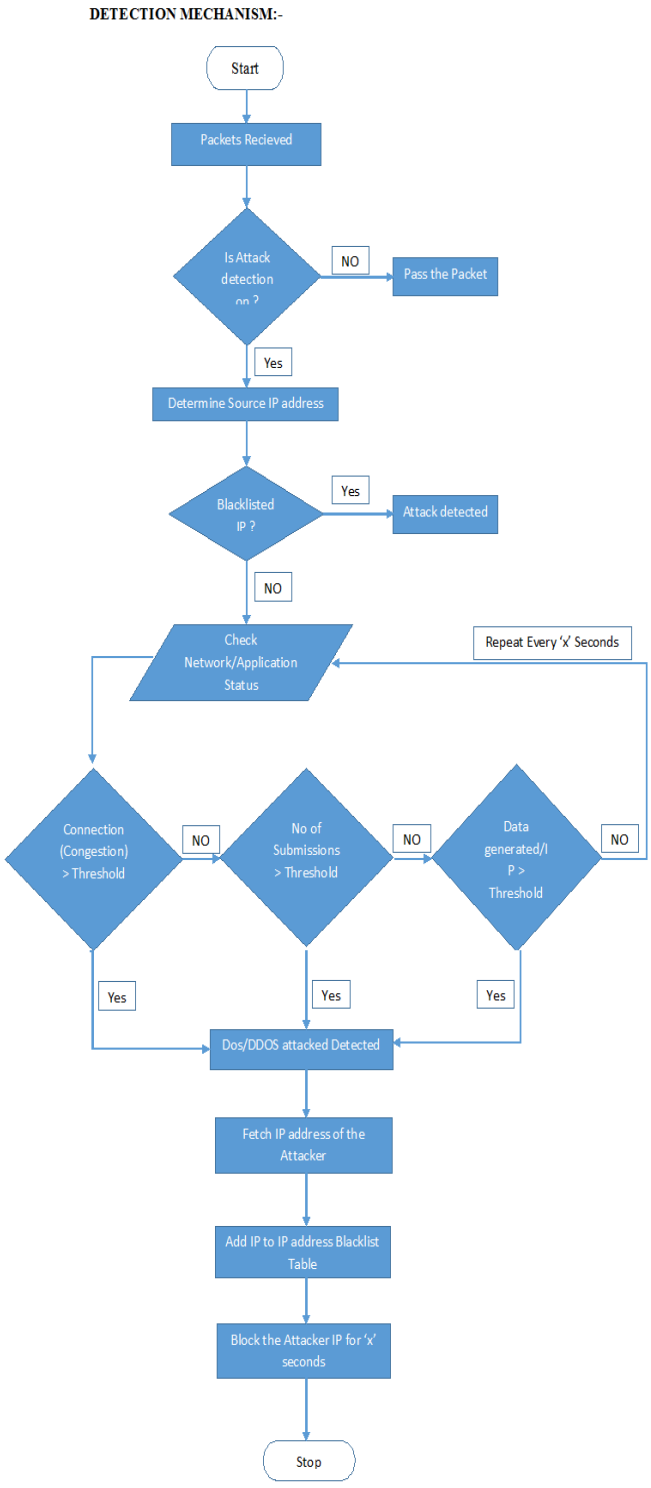
Attack Algorithm

1. repeat
2. t ← 0;
3. while t<=T do
4. nT ← pickRandomTags.(Threshold);
5. tI ← computeInterarrivalTime(CR,nT);
6. sendMessage(nT,tI);
7. t ← t+t_I;
8. end while
9. if !(attackSuccessful) then
10. CR ←(CR +attackIncrement); {Attack intensification}
11. else
12. while !(attack detected) and attackSuccessful do
13. {Service degradation achieved; attack intensity is fixed}
14. nT ← pickRandomTags(tagThresold);
15. tI ← computeInterarrivalTime(CR;nT);
16. sendMessage(nT;tI);
17. end while
18. end if
19. tIM(CR) = computeInterarrivalTime(CR;NT);
20. tIm(CR) = computeInterarrivalTime(CR;1);
21. until(2/(tIM-tIm) < rateThreshold) and ! (attack detected)
22. if attack detected then
23. {Notify to the Master that the attack has been detected}
24. print “Attack detected”;
25. else
26. {Notify to the Master the attack has reached the threshold dT and archived the intensity CR = CRM}
27. print “Threshold reached”;

28. {Continue the attack by using the previous CR value}
29. CR = CR -attackIncrement;
30. loop
31. nT ← pickRandomTags(tagThresold);
32. tI ← computeInterarrivalTime(CR;nT);
33. sendMessage(nT;t I);
34. end loop
35. end if

IX. ENHANCEMENT:-

DETECTION MECHANISM:-

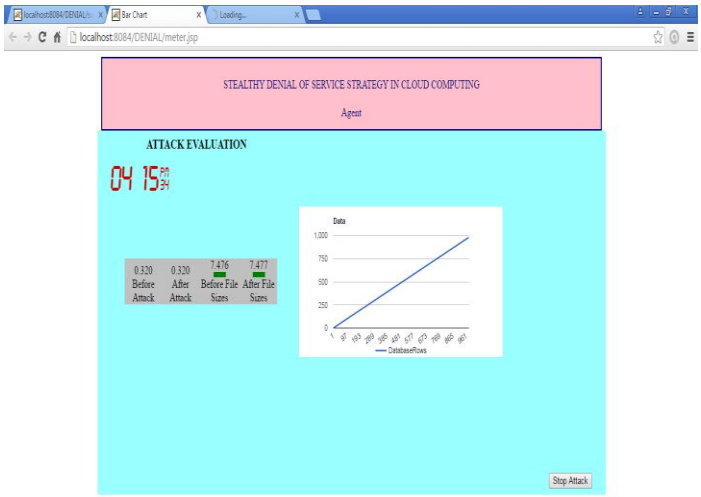


X. CONCLUSION :-

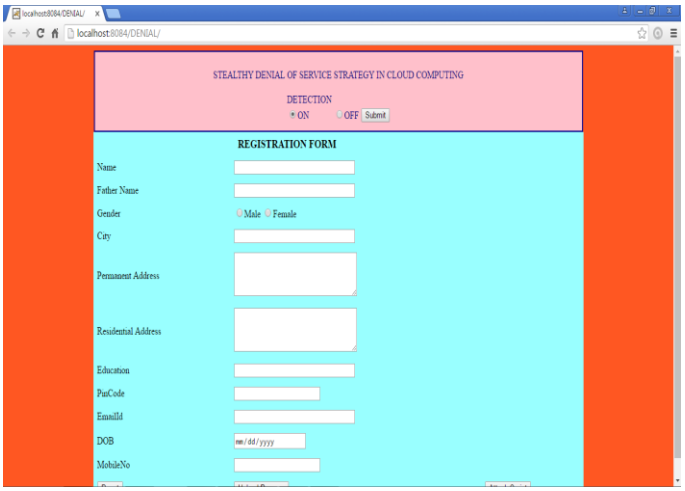
This paper proposes a strategy to implement stealthy attack patterns, which exhibit a slowly-increasing polymorphic behavior that can evade, or however, greatly

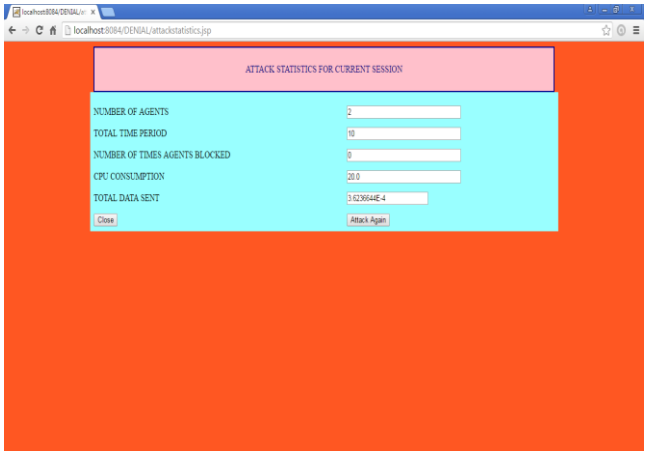
delay the techniques proposed in the literature to detect low-rate attacks. Exploiting a vulnerability[8] of the target application, a patient and intelligent attacker can orchestrate sophisticated flows of messages, indistinguishable from legitimate service requests. In particular, the proposed attack 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.

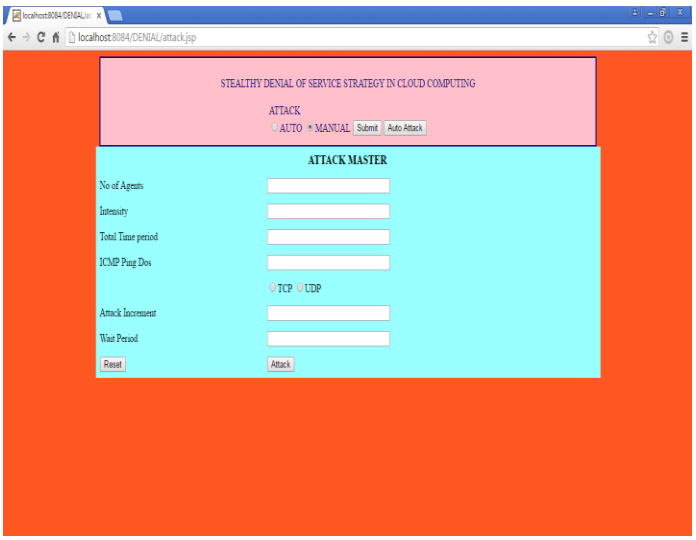
In the future work, we aim at extending the approach to a larger set of application level vulnerabilities, as well as defining a sophisticated method able to detect SIPDAS based attacks in the cloud computing environment.

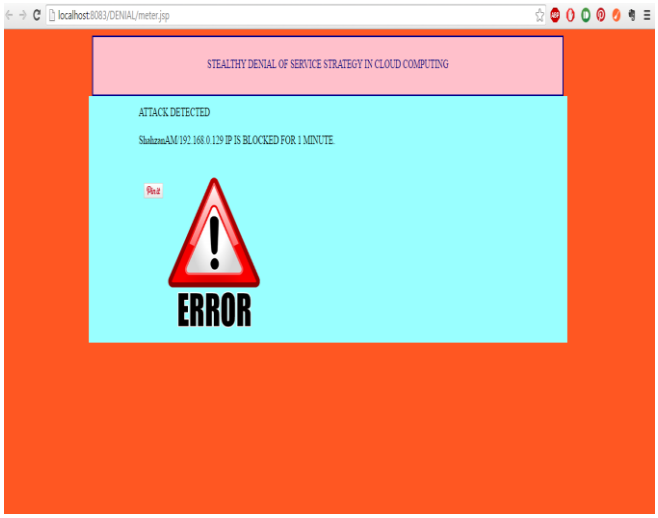


XI. SNAPSHOTS









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]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.