ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

Minimizing Stealthy Denial of Services in Cloud Computing

Dipalee Balu Raut, Prof. Amrit Priyadarshi

PG Scholar, Department of Computer Engineering, Dattakala Faculty of Engineering, Pune, Maharashtra, India

Professor, Department 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.

KEYWORDS: Cloud computing, sophisticated attacks strategy, low-rate attacks, intrusion detection.

I. INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on theInternet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

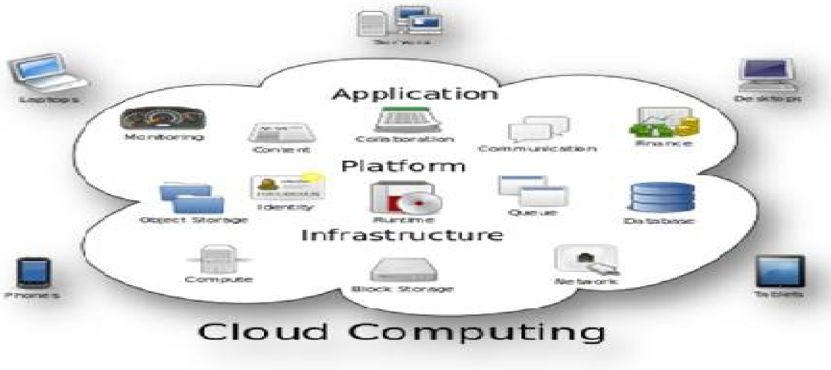


Fig 1: Structure of cloud computing

DOI: 10.15680/IJIRCCE.2017. 0507048

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.

II. LITERATURE SURVEY

Literature survey is the most important step in software development process. BeforeDeveloping the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

- **Security and privacy governance in cloud computing via SLAS and a policy orchestration service.**

AUTHORS: M. C. Mont, K. McCorry, N. Papanikolaou, and S. Pearson.

The 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[2].

- **Security event correlation approach for cloud computing.**

AUTHORS: M. Ficco

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. The escalation process from intrusion symptoms to the identified cause and target of the intrusion isdriven by a knowledge-base represented by an ontology. A prototype implementationof the proposed intrusion detection solution is also presented[3].

- **Monitoring continuous state violation in datacentres: Exploring the time dimension.**

AUTHORS: S. Meng, T. Wang, and L. Liu.

Monitoring global states of an application deployed over distributed nodes becomes prevalent in today's datacentres. State monitoring requires not only correct monitoring results but also minimum communication cost for efficiency and scalabilityFirst, WISE uses distributed filtering time windows and intelligently avoids global information collecting to achieve communication efficiency, while guaranteeing monitoring correctness at the same time.

- **Performance analysis of network I/O workloads in virtualized data centres**

AUTHORS: Y. Mei, L. Liu, and X. Pu

Server consolidation and application consolidation through virtualization are key performance optimizations in cloud based service delivery industry. In this paper, we argue that it is important for both cloud consumers and cloud providers to understand the various factors that may have significant impact on the performance of applications running in a virtualized cloud. This paper presents an extensive performance study of network I/O workloads in a virtualized cloud environment. Finally, we analyse the impact of different CPU resource scheduling strategies and different workload rates on the performance of applications running on different VMs hosted by the same physical machine[5].

III. RELATED WORK

A. Problem Statement:

The Design system Stealthy Denial of Service Strategy in Cloud Computing.The solution to this problem is desperately desired by cyber defenders as the network security community does not yet have solid answers. Different from previous modelling methods, we propose a two layer epidemic model. the upper layer focuses on networks of a large scale networks, for example, domains of the Internet; the lower layer focuses on the hosts of a given network. This two layer model improves the accuracy compared with the available single layer epidemic models in malware modelling. Moreover, the proposed two layer model offers us the distribution of malware in terms of the low layer networks.

B. 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 thatare 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.

IV. PROPOSED ALGORITHM

A. Design Consideration:

Algorithm [A]: SIPDAS Algorithm

```
Require: timeWindow = T(Burst period)
Require: Integer nT = 0 (Nested tags within each message)
Require: Integer tagThresold = NT (Nested tags threshold)
Require: Integer rateThreshold = dT (Attack rate threshold)
Require: Integer attackIncrement = DI (Attack intensity increment)
Require: Integer CR I0 (Initial attack intensity)
repeat
    t( 0;
    while t _ Tdo
        nT( pickRandomTags(tagThresold);
        tI( computeInterarrivalTime(CR; nT);
        sendMessage(nT , tI );
        t ( t + t1;
    end while
    if !(attackSuccessful) then
        CR (CR + attackIncrement);
        Attack intensification
    else
        while !(attack detecte) and attack Successful do
            Service degradation achieved; attack intensity is fixed
            nT( pickRandomTags(tagThresold);
            tI( computeInterarrivalTime(CR; nT);
            sendMessage(nT, tI);
        end while
    end if
    timCR= computeInterarrivalTime(CR,NT );
    timCR= computeInterarrivalTime(CR, 1);
until (2/(tim-tim) less than rateThreshold) and !(attack detected)
if attack detected then
    Notify to the Master that the attack has been detected
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

```
print ’Attack detected’;
else
Notify to the Master the attack has reached the threshold dT and archived the intensity CR 1/4 CRM
print Threshold reached;
Continue the attack by using the previous CR value
CR = CR - attackIncrement;
loop
nT( pickRandomTags(tagThresold);
tim( computeInterarrivalTime(CR, nT );
sendMessage(nT, tI );
end loop
end if
```

Algorithm[B]: Detection Algorithm

```
Start
Compare online Datapackets backup to data packets through XML Analyzer
If both online data packet != backup data packet
Store the difference as the difference between data packets(PacketDiff)
If PacketDiff greater than Threshold
Check the Network Application Status
If != OK then DDOS attack detected
Get the IP address of user who last updated the online data packet
Block that IP address
Add IP to the Blacklisted IP address
Repeat it for every X minutes
End
```

B. Description of The Proposed Algorithm:

SIPDAS: Present a strategy to implement attack patterns that decreases the problem, it is an incremental and iterative process. In first iteration limited number of flows detected. Service degradation is achieved. Algorithm 1 follows the approach to perform a stealthy degradation service in the cloud. Based on the requests and resources, the requests greater than resources, then the file was blocked by the admin or master.

Implementation of SIPDAS attack can be done in several ways. Here, use the same framework assumed for building up the target application server SUA. when the attack is activated by the web, a parameters is sent to the Master, including the URL. The master acquires periodically information from the store and sends messages to agents in order to update their actions, according to attack strategy.

V. PROPOSED SYSTEMS

In proposed system 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 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 customercan 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 applicationvulnerabilities, in order to degrade the service provided by the target application server running in the cloud.

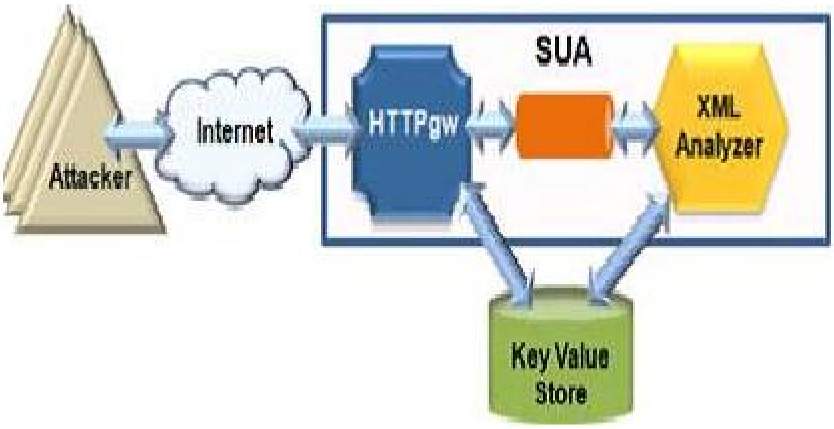


Fig.2. System Architecture.

Advantages of Proposed System:

- We show that the proposed slowly-increasing polymorphic behaviour induces enough overload on the target system 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 or a different timing in order to inflict a prolonged consumption of resources.

VI.MODULES DESCRIPTION

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. 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.

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 customers flows and the DoS flows. Denote C0 as the load at time the onset of an attack period T, 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, a patient and intelligent attacker should be able to discover an application 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.

D. XML-Based DoS Attack:

During the experimental campaign, we analysed 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.

VII. RESULT ANALYSIS

Even if a person harmed , the attack process can be begin by working a different application polymorphism in the form, overtime in order to visit a action continuing for a long time. The performance degradation is achieved through SIPDAS agent, and also service visible instead of invisible by using mOSAIC framework. User has so many problems, implement the user-friendly security, and reduce the cost.

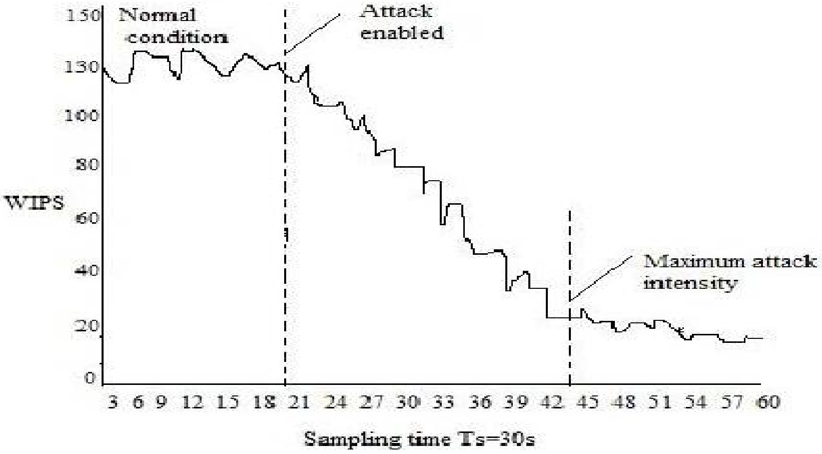


Fig. 3: SIPDAS effect on the mOSAIC based application,

Fig.3. SIPDAS effect on the mOSAIC based application Fig.3 shows the performance degradation by using SIPDAS agent algorithm, the attack intensity increased, the attack successful is true and attack is detected, the admin that maximum average message rate is reached and continue to inject messages formatted. The current agents reached threshold, the master replaces them with new agents, maximum level of intensity achieved by the previous agents. The attack starts with a limited number of agents. (i.e., Single Agent). Auto scaling mechanisms is enabled by the mOSAIC framework.

Admins Get Notification of Attacker:

When Attacker attacks on our system then Admins get automatically notifications and

Admins get IP address of attacker.

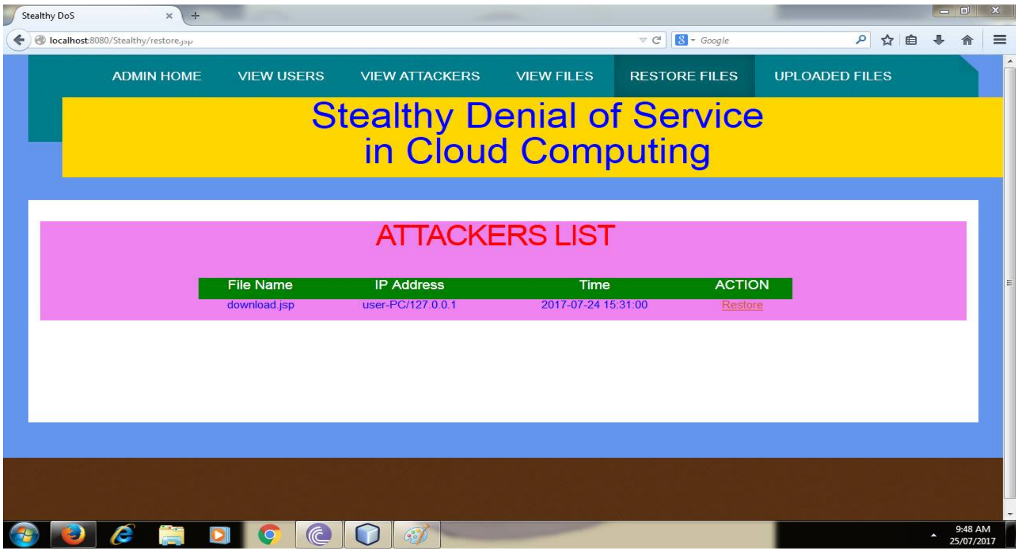
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017



Stealthy Denial of Service in Cloud Computing

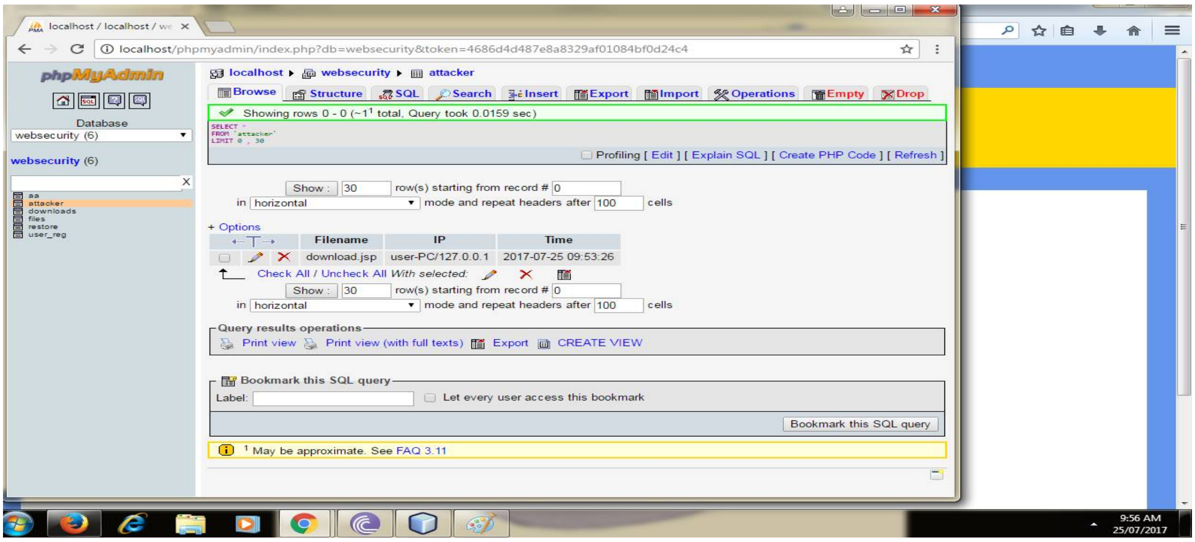
ATTACKERS LIST

File Name	IP Address	Time	ACTION
download.jsp	user-PC/127.0.0.1	2017-07-24 15:31:00	Restore

Fig 4: Snapshot showing Admins get Notification of Attacker

Admin Restore Block File:

When attacker attacks on our system then admins gets IP address of that attacker. When admins get Notifications of attacker then Admin block the IP address of that attacker.



Showing rows 0 - 0 (~1[1] total, Query took 0.0159 sec)

```
SELECT *
FROM `attacker`
LIMIT 0 , 30
```

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

	Filename	IP	Time
□	download.jsp	user-PC/127.0.0.1	2017-07-25 09:53:26

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Label: □ Let every user access this bookmark

Bookmark this SQL query

[1] May be approximate. See FAQ 3.11

Fig. 5: Snapshot showing Admin restoring Block file

VIII. CONCLUSION AND FUTURE WORK

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.

In future work, carrying the approach to large set application level exposure, apart from thisfixing a sophisticated method capable to detect SIPDAS attacks in cloud environment.

ACKNOWLEDGMENT

I would like to express my gratitude towards my guide and H.O.D of Computer Engg. Department Prof. PriyadarshiAmrit 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.