# New Approach to Mitigate XML-DOS and HTTP-DOS Attacks for Cloud Computing

Reza Manouchehri Sarhadi
Research Scholar,
Dept. of Applied Computer Science,
UITM, Rzeszow, Poland,

Vahid Ghafori
Research Scholar,
Dept. of Applied Computer Science,
UITM, Rzeszow, Poland,

## ABSTRACT
Moving towards Cloud Computing is accelerating and businesses are trying to present their software in the cloud. Cloud uses SOA and web services to present always accessible services which raise up threats and vulnerabilities.
Users need to access Cloud from anywhere and this availability comes from presenting services as Web Service over the Internet. Web service in Cloud Computing specially in SaaS plays an important role to present business functionality. Web services are intended to be accessible from different places and applications.
It leads to evolve some vulnerabilities which have to be seriously considered.
One of major vulnerabilities is DDoS attack based on HTTP protocol and XML technology called HTDOS and XDOS which works on layer 7 OSI model and can easily pass through firewalls and take down the server.

In the paper we develop a Cloud defender system called CSQD (Cloud Service Queuing Defender) to detect and mitigate XML vulnerabilities in web services.
CSQD also applies a traceback solution to discover origin of attack.
CSQD system is a self-learner system which means if an attack successfully brings down the server the CSQD finds the malicious request and adds it to its database to stop the same future attacks.
Our results show that CSQD is effective and efficient in detecting and mitigating most of DoS attacks.

## Keywords
Cloud Computing, SaaS , XDoS , HDoS , DDoS.

## 1. INTRODUCTION
Cloud Computing is divided into three models [1]: Infrastructure as a Service (IaaS), Platform as a service (PaaS) and Software as as Service (SaaS).
SaaS has become popular as delivery model which supports Service Oriented Architecture (SOA) and web services technologies for many business applications [2].
Cloud Computing and SOA complete and support each other nevertheless they can be followed separately or simultaneously . SOA permits user frontend applications and enterprise backend servers to have easily approach to cloud offerings by supplying a backbone [11].
Services in SOA mostly implemented in Web Services because they are based as the standard technology [9].
Web Services can act as a communication channel to connect distinctive messaging platform , making information available between applications and publish inside functions over the internet [10].
Users need to access Cloud from anywhere and this availability comes from presenting services as web service over the Internet and causes SaaS layer to have lowest security level.
In SaaS model for public cloud, attacks target following area [3]:

- Availability
- Data Security
- Network Security
- Identity Management

As [4] says availability is considered as one of the top three concerns for CIOs. Occurring a DoS attack which suspends a business for some hours can lead to big losses for the business.
Web Services are vulnerable to various attacks which occurring one can question a business at least.
One of the most typical attacks is DOS attack which can get caught by IDS but when it comes to web service, situations are completely different [12].
According to [14] 94% of data centers have seen DoS attacks and most of attacks have taken placed over HTTP.
Web services are mostly invoked over HTTP protocol and it gives attackers good opportunity to travel through IDS and firewalls. [13]
Using web services in SaaS introduces new DDoS attacks namely HyperText Transport Protocol (HTTP) and Extensible Mark-up Language (XML) Denial of Service (DoS) attack or HX-DoS attack which their aims are to take down a web service or system running that service.
DOS attacks in Web Services are excessively asymmetric. It takes attacker a little to launch a attack payload without wasting a lot of bandwidth or CPU . Besides XML processors are vulnerable to DOS attacks even those already tested [16].

There are some standards for Web Services which ordinarily cover other aspects of security such as confidentiality, authentication and so on but none of them address DoS attacks even some of them are prone to DoS attacks [15].

## 2. Related Work
Many works have been done in the network security area but security for cloud computing is something new and challengeable. In direction of security in cloud computing lots of research is moving towards. Every day is seen that Cloud is having the problem according to new vulnerabilities and

exists several live examples in which cloud is enduring new attacks.

Using web Services in Cloud leads to bring one of the most important attacks to cloud computing which comes from HTTP Denial of Service or XML-Based Denial of Service attacks. These kinds of attacks are easily implemented but several times more difficult to be stopped.

Previous work on cloud security defense was done by Ashley Chonka [4]. He proposed a protector based on neural network to detect and filter DOS attacks and also offered a solution to discover origin of attack in the basis of tracebacking.He had developed the solution according to his previous research on SOTA, which was based on service-oriented architecture and service-oriented grid architecture.

Alwyn Roshan Pais [5] , presented a XML firewall to mitigate different XML vulnerabilities . The firewall is based on Role Based Access Control (RBAC) model and validates the input xml documents before sending to the web services.

Similar work to Chonka was done by Lanjuan Yang [6]. He offered an approach namely SBTA which was in the basis of Service Oriented Architecture (SOA) and suggested a filter approach to filter attacks and a solution to find source of attack.

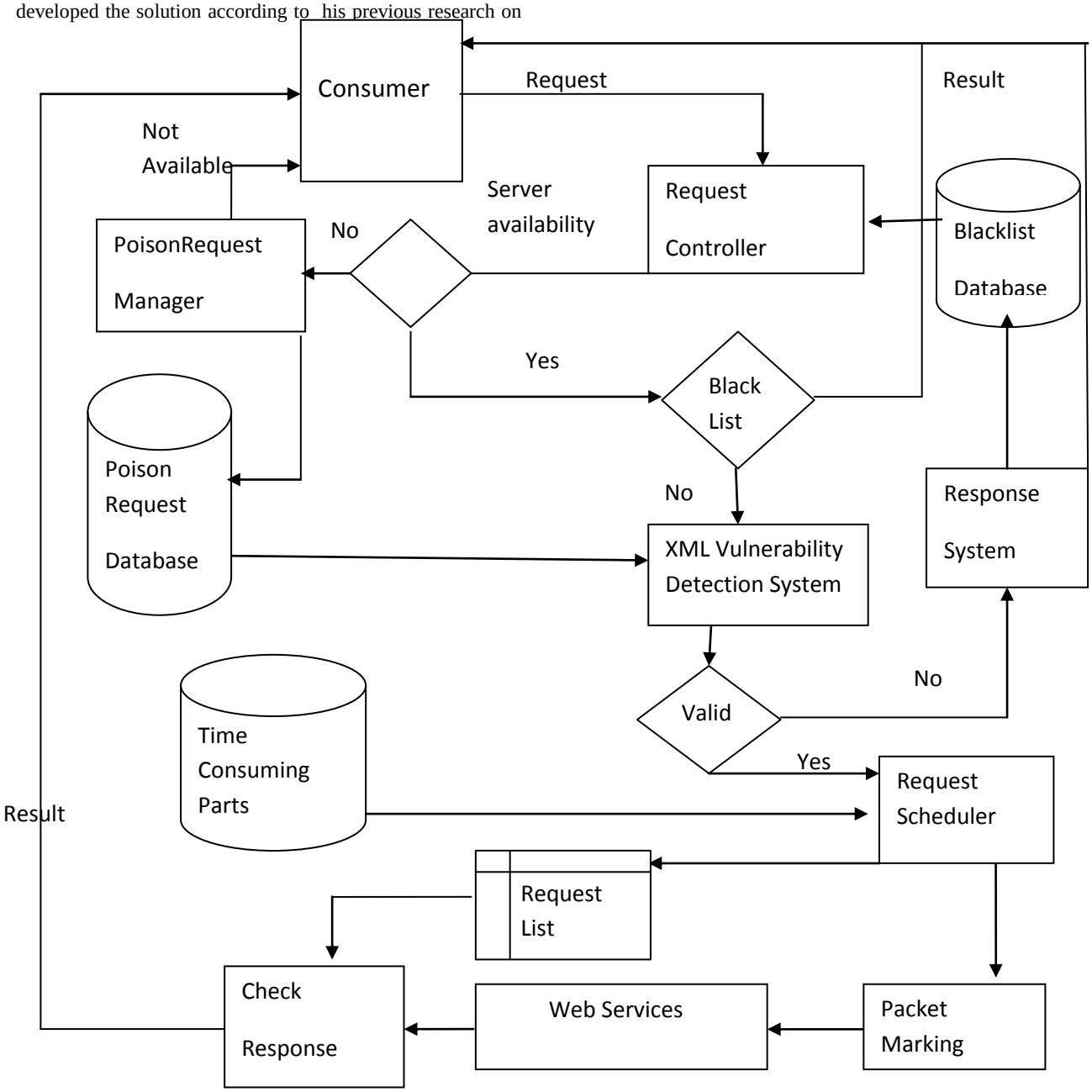


**Figure 1: CSQD Flowchart**

Another work in same area is done by Tarun Karnwal [7]. He offered a security service called filtering tree, which work like a service broker within a SOA model. It is converting the consumer request in XML tree form and uses a virtual Cloud defender which will defend from these types of attacks.

# 3. Cloud Service Queuing Defender (CSQD)

## 3.1 Architecture

CSQD is a detector and defender system which also presents a way to find source of attack.
It can be placed anywhere between server and client but it is suggested to be placed as much as close to ingress router
As Figure1 illustrates CSQD contains following parts:

- Request Controller
- XML Vulnerability Detection System
- Poison Request Manager
- Request Scheduler
- Packet Marking
- Check Response
- System Response

When a request is sent by a consumer it is first checked whether or not the server is up.
It is possible that last request has taken down the server. In this case Poison Request Manager is informed to find last activity which caused server unavailable.
For normal circumstance the request is directed to XML Vulnerability Detection System to check request against different XML attacks such as malicious XML messages, XML DoS attacks. Afterwards the request is forwarded to Request Scheduling If no negative response is received.
In situations that an attack is detected it will be sent to Response System. This system prepares an appropriate message and also inserts the sender‘s IP address into Blacklist database. Request Scheduler puts the request in a list data structure. This data structure is used to track and find poison and malformed requests. If a request is put in the list it will be processed by the server otherwise will be kept in waiting state.
Packet Marking adds some tags in header of request. These headers are not modified during traveling the network. These tags will be used to find source of attack.
After processing the request, the web service forwards the results to Check Response. Check Response validate the response and removes processed request from Request List

## 3.2 Data Stores

We have used three databases and one internal data store in our architecture.

- Blacklist database :
  It contains IP addresses which have to be blocked.
  Once a report comes from detection module which indicates a request is malicious Response system
- Poison Request database :
  The database is used to record discovered attacks. Once the server recovers after an attack system looks for the request which has brought down the server and saves it in the Poison Request database.
  It has three attributes namely IP address, content and date
- Time-consuming parts :
  It contains two attributes namely name and URL.
  Administrator enters these information manually.
- Request List :

It is an internal data store which keeps and tracks incoming requests. Every element of this data store consists five attribute namely IP address, content, requested URL, ID and date.
This data store plays an important role in the system. Requests which are inserted in the Request List will be processed

## 3.3 Algorithm for CSQD

**Step 1**: wait for new request

**Step 2**: Check the server if it is up or down .

**Step 3**: If the server is down find malicious request and go to Step 1

**Step 4:** Accept a new request.

**Step 5:** If the request is in Blacklist discard it and go to Step 1

**Step 6**: Send request to detection module:
Check size of the request
Check DTD content
Check malicious content
Check DOS pattern

If one of the above criteria occurs, send the request to Step 6 otherwise send it to Step 8

**Step 7**: Put the IP address in the Blacklist and send an appropriate message and go to Step 1

**Step 8**: If the request is a type of time-consuming part and threshold for maximum concurrent time-consuming part has reached go to waiting state.

**Step 9:** Generate unique ID for the request and extract some useful information from the request.

**Step 10**: If Request List has enough space insert the request otherwise go to waiting state.

**Step 11**: Add the unique ID and IP address of router and request to the header

**Step 12**: process the request

**Step 13**: Check the response and find the request in Request List and remove it

# 4. Evaluation

In this part we have evaluated our implemented defense system against different type of XML attacks. A Windows Framework Communication (WCF) application is developed in C# to expose different web services. This WCF application is a SOA based application which presents an online music store.
It is also developed a front-end application which consumes web services and provides normal and malformed traffics. Two computers are used as a consumer and supplier. We setup our firewall in the supplier system and front-end application in consumer system.
At first we measure response time for requests without using our defense system. Afterwards we deploy the defense system and calculate the response time again. The time elapsed per request without using defender is shown in Figure 2.
Then we calculate response time with defender. Figure 3 shows a performance for web service using CSQD.

It shows that we have a overhead using our defender. As number of request increases the overhead rises.

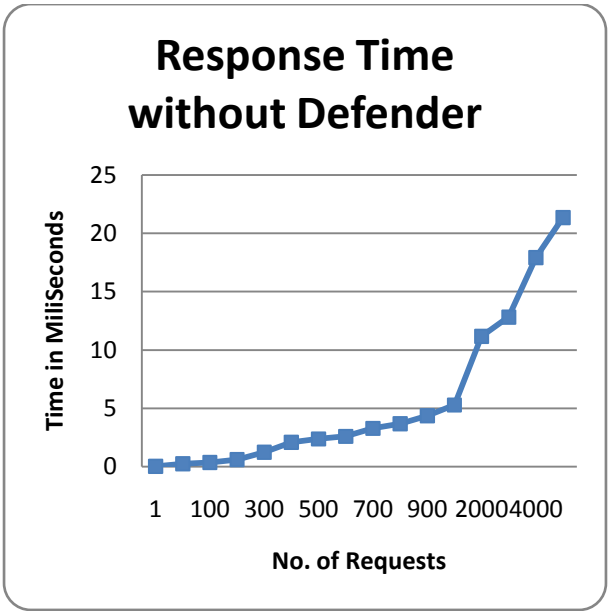


**Figure 2: Response Time without CSQD**

To show this overhead we subtract both previous results and As we can see in figure 4 the difference between two charts. The overhead depends on some criteria:

- Buffer Size:
  Size of buffer is one of the important effective key in performance. If size of buffer is low and number of requests is high the defender uses too much delay to process requests.

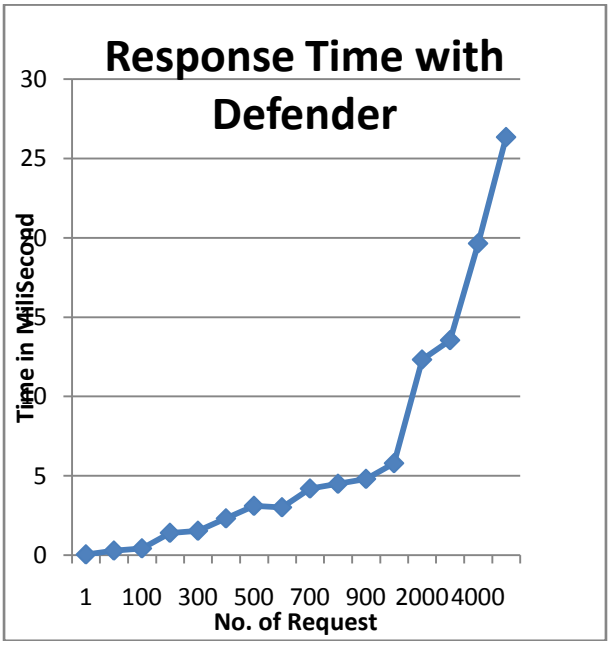


**Figure 3 :Response Time using CSQD**

- Response time :

If a response time for a request is high the request will remain in the buffer more. In other words the request occupies the buffer up to gets a response back. However it prevents entering other request in the buffer and the defender is forced to use delaying.

- Waiting Time :

The time which is given to a request when the buffer is full and the request will be in the waiting state. If a waiting time is high it is possible that buffer finds a free space but the request is still in the waiting state.

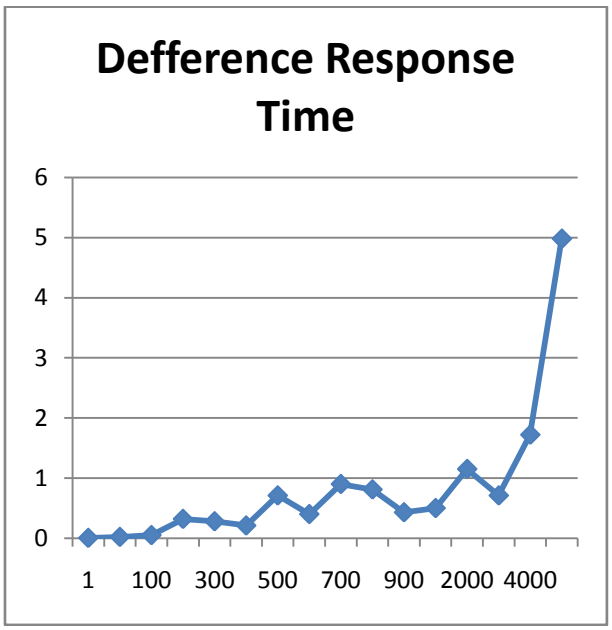


**Figure 4: Difference Response Time**

## 5. CONCLUSION

According to results obtained, our CSQD is effective and efficient in detecting and mitigating most of DOS attacks.
CSQD system is a self-learner system which means if an attack successfully brings down the server the CSQD finds the malicious request and adds it to its database to stop the same future attacks. To achieve this target CSQD keeps requests in a buffer till a response backs to the system.
It also adds some tags in the header of requests to find source of attack because IPv6 will replace IPv4 and current IP traceback will not be supported.

## 6. REFERENCES

[1] Peter Mell, Timothy Grance. The NIST definition of cloud computing. NIST. [Online] September 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[2] definition Software as a Service (SaaS). SearchCloudComputing. [Online] http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service.

[3] Clinton DSouza, Rafael Santana. Vulnerabilities in SaaS Layer of Cloud Computing. s.l. : Arizona State University, 2012.

[4] Page, Scott. Cloud Computing-Availability. SlideShare. [Online] http://www.slideshare.net/s2page/cloud-computing-availability-8517731.

[5] Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. Ashley

Chonka, Yang Xiang n, Wanlei Zhou, Alessio Bonti. 2011, Elsevier, pp. 1097–1107.

[6] Protection against Denial of Service and Input Manipulation Vulnerabilities in Service Oriented Architecture. Alwyn Roshan Pais, Deepak D.J., and B.R. Chandavarkar. Chennai ,India : Springer, 2011. Advances in Network Security and Applications. pp. 331–343.

[7] Defense of DDoS Attack for Cloud Computing. Lanjuan Yang, Tao Zhang, Jinyu Song, JinShuangWang, Ping Chen. Zhangjiajie, China : IEEE, 2012. Computer Science and Automation Engineering (CSAE). pp. 626-629.

[8] A Comber Approach to Protect Cloud Computing against XML DDoS and HTTP DDoS attack. Tarun Karnwal, T. Sivakumar, G. Aghila. Bhopal : IEEE, 2012. Electrical, Electronics and Computer Science (SCEECS). pp. 1-5.

[9] T. Erl, Service-Oriented Architecture (SOA): Concepts, Technology, and Design, Prentice Hall, 2005

[10] J. B. ,. A. G. Rajkumar Buyya, Cloud Computing: Principles and Paradigms, Hoboken, New Jersey: John Wiley & Sons , Inc, 2011.

[11] F. Bowen, "How SOA can ease your move to cloud computing," [Online]. Available: http://www-01.ibm.com/software/solutions/soa/newsletter/nov09/article_soaandcloud.html.

[12] CCNA Security Course booklet version1.0, Indianapolis: Cisco Press, 2010

[13] M. Harwood, Security Strategies in Web Applications and Social Networking, Jones & Bartlett Learning,LLC, 2011

[14] P. Dinham, "Denial-of service attacks vulnerability increases with the cloud," 29 Januaury 2013. [Online]. Available: http://www.itwire.com/business-it-news/security/58480-denial-of-service-attacks-vulnerability-increases-with-the-cloud.

[15] Elisa Bertino, Lorenzo D. Martino , Federica Paci ,Anna C. Squicciarini. 2010. Security for Web Services and Service-Oriented Architectures. s.l. : Springer, 2010.

[16] Harwood, Mike. 2011. Security Strategies in Web Applications and Social Networking. s.l. : Jones & Bartlett Learning,LLC, 2011.