Chonka, Ashley and Zhou, Wanlei 2009, Defending grid web services from XDoS attacks by SOTA*, in Percom 2009 : Proceedings of the Seventh Annual IEEE International Conference on Pervasive Computing and Communications*, IEEE Computer Society, Piscataway, N. J., pp. 1-6.

Defending Grid Web Services from XDoS Attacks by SOTA

Ashley Chonka, *Member, IEEE* and Wanlei Zhou, *Member, IEEE*,
School of Engineering & Information Technology
Deakin University
Geelong, 3220, Australia

Yang Xiang, *Member, IEEE*
School of Management and Information Systems
Central Queensland University
Rockhampton, 4702, Australia

{ashley, wanlei}@deakin.edu.au and y.xiang@cqu.edu.au

Abstract — Grid Web Services are still relevantly a new to business systems, and as more systems are being attached to it, any threat to it could bring collapse and huge harm. Some of these potential threats to Grid Web services come in a new form of a new denial of service attack (DoS), called XML Denial of Service or XDOS attacks. Though, as yet, there have not been any reported attacks from the media, we have observed these attacks are actually far less complex to implement than any previous Denial of Service (DoS), but still just as affective. Current security applications for grid web services (WS-Security for example), based on our observations, and are not up to job of handling the problem. In this paper, we build on our previous work called Service Oriented Traceback Architecture (SOTA), and apply our model to Grid Networks that employ web services. We further introduce a filter defence system, called XDetector, to work in combination with SOTA. Our results show that SOTA in conjunction with XDetector makes for an effective defence against XDoS attacks and upcoming DXDoS.

Index Terms — Grid Web Service, Service-Oriented Computing, SOTA, XDectector

I. Introduction

The foundation for implementing a Service-Oriented Architecture (SOA) onto a grid network can be found in Service-Oriented Grid Computing (SOGC). It is mostly agreed upon, that Open Grid Services Architecture (OGSA) [1][2][3] is the cornerstone of SOGC, which describes itself as, a high level architecture abstraction of service–oriented grid computing, while Web Service Resource Framework (WSRF) [4] is the specification for implementing the OGSA model. In the area of Security, OGSA employs WS-Security [8], XML-Signature [9], XML-Encryption [10] to encompass message integrity, confidentiality and availability [6][7]. These standards work in conjunction with Simple Object Access Protocol (SOAP) [11], regardless of the transport protocol being used.

The SOA model is currently adopted in various areas of distributed application development, for example design and implementing web applications. In the area of OGSA, SOA is mostly used by large scale corporations to help find, and share service resources, internally. It is, also, sometimes extended for external use, in order to expose corporate resources to the outside world. The results from this type of implementation, has earned corporations, billions of dollars in revenue. Though one of the downsides of this implementations, is they are totally dependent upon the SOA models, for their income. Any problems or threats, such as a Denial of Service (DoS) attack for example [13][14], to these services, would result in billion dollar losses and maybe corporate collapse.

Probably the weakest part of the OGSA security, based on our observations, is message delivery and integrity, while security focus has been on message protection [12]. With this weakness, a gap has opened up and a new series of attacks are now available for those who would like to implement them. In fact, we could see headlines about these new attacks in the coming days and months. These new attacks have been

978-1-4244-3304-9/09/$25.00 ©2009 IEEE

referred by researchers as XDoS (XML based DoS) and DXDoS (Distributed XML-based DoS) [21].

The papers by Jenson et al. [6] and Padmanabhuni et al [21] discusses the depth of this problem by showing some of the effects of these attacks, and discuss how one might be able to prevent them. It should be noted at this time, we use the term DXDoS and XDDoS [21] as inter-changeable terms. One of the many reasons why an attacker would chose this new form of Denial of Service (XDoS) attack over previous DoS, is due to how much simpler and devastating these attacks are against Web Services. Secondly, there has been a lot of research on developing defense systems [13][14] against DoS and DDoS attacks, in order to prevent them. With XDoS, and soon DXDoS, these attacks are relevantly new, so counter measures or counter systems are under developed and in effect non-existent.

In this paper, we follow on from our previous work [15][16] on Service-Oriented Traceback Architecture (SOTA), by applying our framework to OGSA. We further add to our work by introducing a defense filter called XDetector [XML Detector], in which it is distributed throughout the grid, in order to properly defend it. Our system is one of the first defense systems to attempt to defend against these new attacks, though we should point out that at the time of producing this paper, XDetector was trained and tested on the MIT Dataset [17], since no dataset with the new attacks has yet been produced, though we are in the works on developing such a set. The remainder of the paper is made up of the following: Section 2 reviews the related work on SOTA. Section 3 covers the details of applying our SOTA framework, and XDetector, to OGSA. Section 4 presents our experiments and performance evaluation. Lastly, Section 5 provides our conclusions and future research.

II. Related Work

In this section, we discuss very briefly Service-Oriented Traceback Architecture (SOTA) and the new forms of Denial of Service Attacks called XDoS.

A. Service-Oriented Traceback Architecture

SOTA is a web security service application that is product-neutral. Its main objective is to apply a SOA approach to traceback methodology. This is in order to identify a forged message id, since one of the main objectives of XDoS and DXDoS is hide the attacker's true identity.

The basis of SOTA is founded upon the Deterministic Packet Marking (DPM) [17] algorithm. DPM marks the ID field and reserved flag within the IP header. As each incoming packet enters the edge ingress router it is marked. The marked packets will remain unchanged as they traverse the network. Outgoing packets are ignored.

DPM methodology is applied to our SOTA framework, by placing the Service-Oriented Traceback Mark (SOTM) within web service messages. If any other web security services (WS-Security for example) are already being employed, SOTM would replace the 'token' that contains the client identification. Real source message identification are stored within SOTM, and placed inside the SOAP message. SOTM, as in DPM tag, will not change as it traverses through the network. The composition of SOTM is made up of one XML tag, so not to weigh down the message, and stored within a SOAP header. Upon discovery of an XDos or DXDoS attack, SOTM can be used to identify the true source of forged messages.

SOTA does not directly eliminate an XDoS or DXDoS attack message. This is left for the filter section of a defense system (Firewalls or our new filter XDetector). Instead SOTA's two main goal is to deal with the two main objectives of XDoS, which are: exploit a known vulnerability, in order to bring down system. These vulnerabilities could be found in communication channels (flooding for example) or known exploits within the services provided (for example, an attacker can Overload their messages, which will result in the web server crashing). The second objective is that attackers try to hide their identity. The reasons vary, depending on what type of attack, but usually it is to cover their crime or to bypass a known defense that is in place to prevent it. It is with this second objective that SOTA attempts to cover, as other traceback methods, like Probability Packet Marking (PPM) [18] and DPM.

There are many reasons for OGSA to employ a SOTA type framework, some of these are:

- Current web security is not up to handling an XDoS or DXDoS attack. In fact, as Jension et al. shows how WS-Security can be used in an XDoS attack.
- SOTA does not violate IP protocols, in order to store information for traceback purposes.
- Using the SOA model, SOTA can be employed on any ubiquitous grid system.

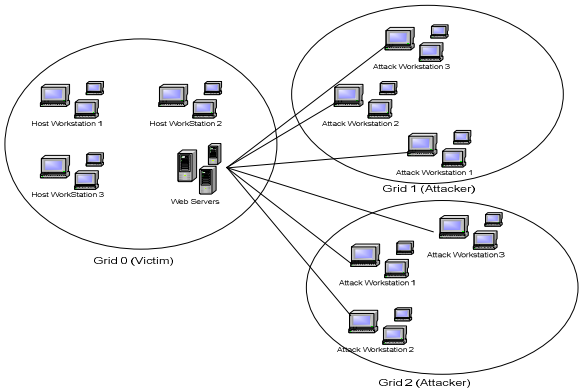

Figure 1. Distributed XML-based Denial of Service attack, where an attacker has taken control of 2 Grids, and sending huge amounts XML-based traffic to Grid 0 Web Server.

- With IPv6 coming into fruition [19], current IP traceback methods will no longer be viable. This is due to the changes that IPv6 introduces, such as, IPSec and the packet header format no longer holds support the fields that are required for IP traceback.

B. XML-Based Denial of Service (XDoS) Attacks

XDoS was a term coined by Padmanabhuni et al. [21], where web services are prone to XML based Denial of Service (XDoS). A Denial of Service (DoS) is where an attacker attempts to deprive legitimate users of their resources [20]. XDoS attack(s), according to Padmanabhuni et al, Jenson et. al. and Chonka et. al. can affect the following area: Firstly, a network can be flooded with XML messages (instead of packets), in order to prevent legitimate users to network communication. Secondly, if the attacker floods the web server with XML requests, it will affect the availability of these web services. Lastly, attackers manipulate the message content, in order to cause the web server to crash. The experiments carried out by Jenson et al focuses on the last point, though they do not call their attacks XDoS.

To adept XDoS into a Distributed Denial of Service paradigm, called Distributed XML based Denial of Service (DXDoS), the attacker uses multiple hosts to attack the victim with XDoS attacks (Figure 1). This type of attack, though none have been reported as yet, will be probably be the most serious threat to OGSA.

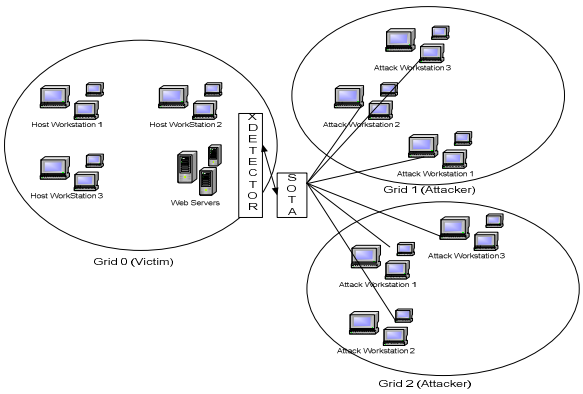

Figure 2. Distributed XML-based Denial of Service attack, where SOTA and XDectector are located just between the each Grid Web Service, in order to detect and filter XDoS attacks.

II. XDoS Defense System for Open Grid Services Architecture

A. Introduction

Inherent internet characteristic, which comprise of limited and consumable resources, is one of the reasons why attackers are so successful. They can target bandwidth, processing power and storages capacities of a grid network. OGSA has limited resources, in order to provide services, which can be exhausted with a sufficient number of consumers. With this particular knowledge, attackers can instigate an XDoS or DXDoS. For example, an attacker can keep sending oversize messages to the web server over a period time. This will result in the web server to crash, since currently there is no restrictions to the size of message [6][21]. In a DXDoS attack, the attacker would order their agents/zombies to instigate a flood attack of oversize messages, against the web server. This again, would result in the web server to crash from either executing the oversize messages, or from communication congestion created from the flood

B. Distributed Defense System

Figure 2 shows our defense system to protect OGSA from XDoS and DXDoS attacks

C. XDos – Detector (X-Dectector)

The XML-Based Detector is trained Back Propagation Neural Network, in order to detect and filter out XDoS messages. A neural network is a set of

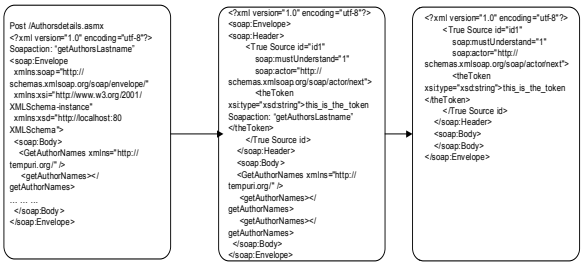

Figure 3. Event Descriptor Graph for SOAPaction attack (Spoofed SOAPAction message (A), SOTM tag (B), True identity of the message, requested by the service provider (C).

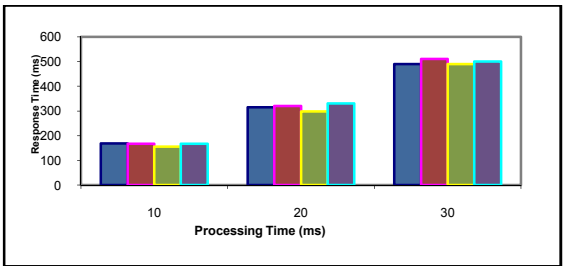

Figure 4. Messages generated by our first simulation (SOAPAction attack)

connected units made up of an input, hidden and output layers. Each of the connections in a neuron network has a weight associated with it. In understanding a neural net is to focus on the threshold logic unit (TLU). The TLU inserts input objects into an array of weighted quantities, sums them up to see if it equals or surpass the set threshold, than outputs the quantity.

$$p = \sum_{n=1}^{n} Th + (w^i x^i) \qquad (1)$$

The threshold is called theta and output is represented by p. It should be noted that a threshold is sometimes called a bias. When Th >=theta than p =1 otherwise p=0. In our observation we use the output layer to represent 0 as legitimate traffic and 1 as attack traffic. The XDectector is located just before the Web Servers, or could be stored on the web server (See Figure 2), in order to provide the greatest resource efficiency and protection.

IV. Performance Evaluation

A. Performance Analysis

In our performance evaluations we show to lots of experiments that we did. Firstly, we have selected two

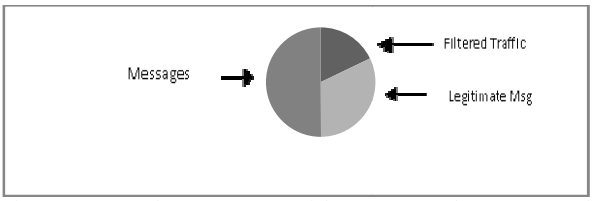

Figure 5. 5 attack messages, out of the 20 generated, were removed after traceback and filter protocols (SOAPAction attack).

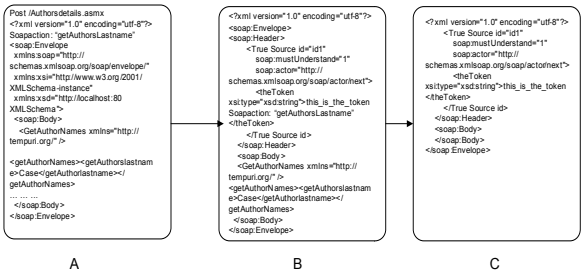

Figure 6. Event Descriptor Graph for XML injection attack (Spoofed XML message (A), SOTM tag (B), True identity of the message, requested by the service provider (C).

performance experiments that we have previous published in [14][15] on SOTA. We do this to show how effective XDoS attacks are, and how SOTA can be used to detect and traceback the source of the message. The two attacks we simulated were a SOAPaction spoof attacks and XML injection attack. In our second experiment, we show how XDectector would work against an DDoS attack.

B. Experimentation

The first simulation that we conducted was a spoofed SOAPaction attack. It intention is to invoke an operation that is different within the SOAP body, and usually results in a web server crash. Figure 9a displays our spoofed SOAPAction message used in this simulation. The message contains within the SOAPAction the author's first name, but only the author's last name is within the SOAP body. This message composition could result in the server behaving erratically or crashing it.

Figure 4 displays the messages that our simulation generated. As each message passed through SOTA it was marked with a SOTM tag (see figure 3b). Further, we can see from figure 4 that the message stops at msg3, this means that an attack was successful. The service provider will instigate the following procedures: Restart the system, search SOTA reconstruction for the true source of the attack (see figure 3c), and instigate filtering protocols (figure 5).

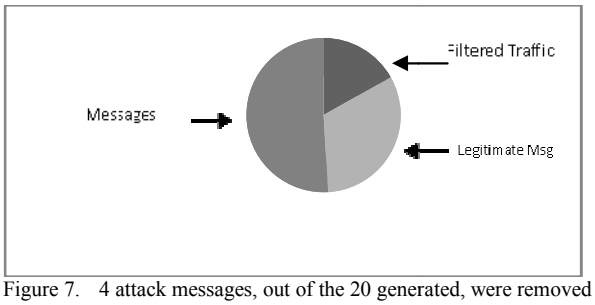

Figure 7. 4 attack messages, out of the 20 generated, were removed after traceback and filter protocols (XML Injection attack).

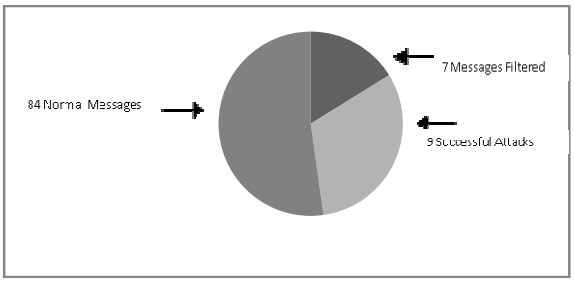

Figure 8. 84 normal messages were processed. 9 floods were successful in crashing the system. 7 attacks message were filtered.

To simulate these procedures, we restarted the program to generate 20 more messages. With the traceback and filtering controls in place, we found 5 attack and 15 normal messages (figure 5).

An XML Injection attack was our last simulation of the 3 XDoS chosen. This attack tries to modify the XML structure of our SOAP message. Figure 6a shows that the authorname tag has another tag within it called authorlastname. The result of this message could lead to a server crash, though it is unlikely. Instead, as shown in figure 6a, the content has been changed. This content change, would lead to incorrect information, being displayed from the tag. Figure 7 displays the messages that our simulation generated. As each message passed through SOTA it was marked with a SOTM tag (Figure 6b). The result of the XML injection attack, shown in Figure 7, is that 4 attack messages were filtered. The first attack message signaled the service provider to instigate SOTA reconstruction. With the discovery of the attacker id, the service provider was able to filter out the rest of the attack messages.

We, also, conducted a message flood attack, using XML Injection. The simulation program was setup to generate a total of 100 messages. If one of those messages was an attack, it had 50/50 chance to crash the system. If the system did crash, a number between 100 and 300 ms was added to the next lot of response time. This was to simulate the time taken by the service provider to restart their system, locate the source, and filter it. From our results, we got 84 normal messages. Further, was the unusually high, 9 successful attacks that crashed the system. The reason for the crashes was due to the chance nature built within our code. These successful attacks are displayed by the groupings within figure 14. Of the attacks that got filtered, 7 attacks messages were discovered.

In our second experiment we trained up XDetector, which contains our Back Propagation Neural Network, in order to detect and filter XDoS attack traffic. In order to train up our Neural Network we used dataset from the week 2, 1998 DARPA intrusion detection evaluation set at Lincoln Laboratory, MIT [17]. The data sets from MIT come in TCP dump format, so we extracted the features we needed and insert them into a MySQL database. These features included SrcIP, DestIP, SrcPort, DestPort and the length of time. We added an extra field to the table for the decision, 0 for legitimate and 1 for illegitimate. Lastly, the main reason to use the MIT dataset is due to fact that an XDoS dataset does not exist at the current time. It should be noted that we at Deakin University are currently working on a dataset, in order to provide this needed dataset to researchers.

Our neural network was trained on a 4 Neuron Layers (3,3,3,1), Learning Rate of 0.2, Momentum of 0.6, and a variable threshold of 0.1 to 0.9, which was incremental increased by 0.1. To show how accurate our neural network is at filtering this DDoS attack, we selected to show sensitivity (Formula 2) and false positive (formula 3).

Overall, our Neural Network performed well, with the sensitivity range of 88% to 94%, with a false positive range 0.05% to 0.45%. Based our simulations, our conclusion is that with any large introduction of network traffic (such as a DDoS flooding attack) will alter the network phase space graph (Figure 10).

$$S = \frac{D}{T_D} \quad (2)$$

$$FPR = \frac{ND}{T_{ND}} \quad (3)$$

Sensitivity (S) is shown by the number of detected attack packets (D) over the total number of packet. False positive rate (FPR) is measured by how many attack packets were detected as normal packets (ND) over the total number of attack packets. Figure 10, confirms that detection of attack traffic is quite early,

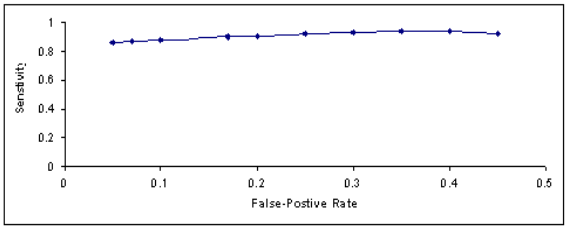

Figure 9. **Sensitivity of Neural Network**

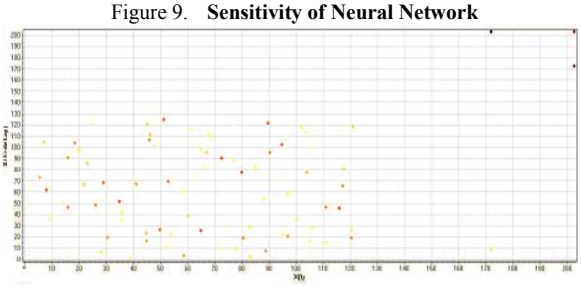

Figure 10. **Filtered DDoS attack traffic from our Neural Network**

in that as soon as the traffic starts to deviate it takes the neural network a few moments to begin to filter the attack traffic.

V. Conclusion and Future Work

This paper builds upon our previous paper [14][15], in which identifies the real source of XDoS attack messages, and filters in order to protect Grid Web Services. SOTA is a traceback system that is constructed on the basis of Web Services. Loose Coupling, Policy Based, Message Based and Dynamic discovery are some of criteria employed by the SOTA framework. XDetector, is a Back Propagation Neural Network, trained to detect and filter XDoS attack message.

The empirical data from our experiments shows that SOTA is efficient and effective. The experimental data also shows that SOTA is able to traceback to the source. Once an attack has been discovered and the attacker's identity known, XDetector can filter out these attack messages. The people, who will be interested in this research, are those that want to their protect web services in a cheap and efficient manner.

Currently, we are planning to move our research on to the Deakin's Enterprise Grid. This will give us access to real-time data and see what the advantages and disadvantages of SOTA and XDectector are.

VI. References

[1] Foster, I., Kesselman, C., and Tuecke, S (2001). "The Anatomy of the Grid: Enabling Scalable Virtual Organizations." Int'l J. High-Performance Computing Applications, vol. 15(3): 200-222.

[2] Foster, I., Kesselman, C., Nick, JM., Tuecke, S (2002). "Grid Services for Distributed System Integration." IEEE Computer 35(6): 37-46.

[3] Foster, I. A. K., C (1999). "The Grid: Blueprint for a New Computing Infrastructure." Morgan Kaufmann San Francisco**,** 1999

[4] WSRF, 2004, Web Services Resource Framework 1.2, December, http://www.oasis-open.org/committees/wsrf/

[5] Nadalin, A., Kaler, C., Monzillo, R., and Hallam-Baker. P., (2008), 'Web Services Security: SOAP Message Security 1.1 (WSSecurity 2004)', http://docs.oasis-open.org/wss/v1.1/, 2008.

[6] Jensen, M., Gruschka, N., Herkenh¨oner, R., and Luttenberger, N., (2007), " SOA and Web Services: New Technologies, New Standards – New Attacks" Fifth European Conference on Web Services, 0-7695-3044-3/07, 2007.

[7] Bishop, M, 'Computer Security', Addison Wesley, 2003

[8] Nadalin, A., Kaler, C., Monzillo, R., and Hallam-Baker. P., (2008), 'Web Services Security: SOAP Message Security 1.1 (WSSecurity 2004)', http://docs.oasis-open.org/wss/v1.1/, 2008.

[9] XML –Signature, (2008), 'XML-Signature Syntax and Processing' http://www.w3.org/TR/xmldsig-core/

[10] XML- Encryption, (2008), 'XML-Signature Syntax and Processing' http://www.w3.org/TR/xmlenc-core/

[11] SOAP 1.1, (2008), http://www.w3.org/TR/soap/

[12] Secure Socket Layer (SSL), (2008), http://en.wikipedia.org/wiki/Secure_Sockets_Layer

[13] Bouzida, Y.; Cuppens, F.; Gombault, S., (2006), '*Detecting and Reacting against Distributed Denial of Service Attacks*' Communications, 2006 IEEE International Conference on Volume 5, June 2006.

[14] Trostle, J, (2006), '*Protecting Against Distributed Denial of Service (DDoS) Attacks Using Distributed Filtering*', Securecomm and Workshops, 2006 Aug. 28 2006-Sept. 1 2006 Page(s):1 – 11.

[15] Chonka,A., Zhou, W., and Xiang, Y., (2008), "Protecting Web Services with Service Oriented Traceback Architecture", *IEEE 8th International Conference on Computer and Information Technology*, IEEE, 2008.

[16] Chonka, A., Zhou, W., and Xiang, Y., (2008), " Protecting Web Services from DDoS attacks by SOTA" *IEEE* 5th International Conference on Information Technology and Applications, IEEE, 2008

[17] Belenky, A.,and Ansari, N., '*Tracing Multiple Attackers with Deterministic Packet Marking (DPM)'*, Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing.

[18] Savage, S., Wetherall, D., Karlin, A., and Anderson, T., (2001), '*Practical Network Support for IP Traceback*', SIGCOMM'00, Stockholm, Sweden, 2000

[19]van Beignum, I, (2008), '*IPv6: coming to a root server near you*', ars technical, http://arstechnica.com/news.ars/post/20080102-icann-to-add-ipv6-addresses-for-root-dns-servers.html , 02 January, 2008

[20] Rogers, L., (2008), 'What is a Distributed Denial of Service (DDoS) Attack and What Can I Do About It?' Computer Emergency Response Team, http://www.cert.org/homeusers/ddos.html

[21] Padmanabhuni, S.; Singh, V.; Senthil kumar, K.M.; Chatterjee,A.Web Services, 2006, "*Preventing Service Oriented Denial of Service (PreSODoS): A Proposed Approach*", ICWS apos;06. International Conference on Volume , Issue , Sept. 2006 Page(s):577 – 584

[22] MIT 1998 DARPA Intrusion Detection Evaluation Data