Web Service Intrusion Detection Using XML Similarity Classification and WSDL Description

Mahdi Bazarganigilani, Belinda Fridey, Ali Syed

Faculty of Business, Charles Sturt University
Melbourne, Australia.
mahdi62b @yahoo.com, bfridey@studygroup.com, asyed@studygroup.com

Abstract

This study introduces a new method to detect the Intrusion attacks on web services using XML similarity classifiers. First, pre-process methods are employed to distinguish the requests both based on the structure and content. WSDL description of the web service is employed to determine the valid structure of a request. A sequential threshold method is employed to define a request as an intrusion. Moreover, an optimized semantic similarity algorithm is employed to define the discrepancy between the contest of the request and the database of normal requests. Evaluation results are conducted based on various methods and algorithm which were presented in academia. The final section presents the promising way with the highest accuracy.

Keywords:. *XML Classification; SOAP Attacks; Webservice IDS.*

1. Introduction

The advances in web technologies that provide attractive features such as ease of use, platform independence, interoperability and so on, has lead to more and more applications such as e-commerce and business intelligence (BI) applications being deployed over the Web Service (WS) platform. Unfortunately, these basic building blocks for WS technology such as XML (eXtended Markup Language), SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery, and Integration), WSDL (Web Services Description Language) due to their inherent security weaknesses, pose vulnerabilities and are the targets for attacks. For example, UDDI is supposed to assist genuine users to locate the appropriate WS, but at the same time also gives chance to attacker to find targets for launching attacks.

WSDL is a document that contains all the information and parameters needed for attackers to launch an attack by submitting malicious code or special characters in order to gain unauthorized access [1]. Although WS-Security Standards [2] are important to address the issues of cryptography, authorization, and authentication they are not useful in eliminating attacks such as SQL/XML injection pertaining to web service-based applications.

This study introduces an Intrusion Detection System (IDS). In each step, the request examined against invalid attacks. A request determined valid if it passes all the steps. A serial multiple thresholds method is employed to examine requests in each step.

2. WS Different Kinds of Attacks and Vulnerabilities

Attackers make use of WSDL description of the web service to understand the various parameters of the victim service. The WSDL document implies the information

which eases the attackers to sieve through document and plan their malicious codes. They use especial characters to submit the malicious codes to obtain the authority to the system or plan a Denial of Service (DOS) attack. There are different kinds of attacks. Attackers may use Recursive payloads attacks, the capability of XML by creating a document with thousands of elements enough deep to break the parser. Moreover, they may oversize the XML request to break the service. To access and manipulate databases, Structured Query Language (SQL) and commands such as Insert, Select, Update, Delete, Drop, Alter and Create are used. In SQL injection, the attacker is able to execute multiple commands in an input field of a SOAP message by using separators like ';' or pipes. By doing so, the attacker is able to gain access to databases, execute stored procedures or invalidated SQL command [3].

As an emerging class of attack, XML injection attack occurs when user input is passed to the XML stream. The XML document is then parsed by the second-tier application or database (DB), the malicious code is then injected into the DB. When it is next retrieved from the DB, it becomes part of the XML stream. In another kinds of attacks, XML parser usually strips "<" or ">" symbols, but the CDATA field allows these non-legal characters to be included in XML document. As a result, after parsing, the CDATA component will be stripped and the resulting string may contain dangerous and malicious characters. Consequently, CDATA field attack may also lead to XML/SQL injection attacks [1].

3. Pre-Process Validation

Various attacks are originated from malicious codes. The major attcks include SQL/XML Injection. To avoid such attacks, each request/response is pre-processed to find any special characters. Figure 1, shows a SQL Injection attack. To protect the WS from such attacks , special characters are extracted in the body of each request [4].

- If a request/response contains '' in the message string, it is blocked.
- If a request/response contains '%' in the message string, it is blocked.

By blocking such messages, many suspicious and malicious codes are prevented.

```
POST /soap/servlet/rpcrouter HTTP/1.0
Host: localhost:9000
Content-Type: text/xml; charset=utf-8
Content-Length: 389
SOAPAction: ""
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsc
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
    <fn:PerformFunction xmlns:fn=" ">
        <fn:uid>8123</fn:uid>
            <fn:password>
                'or 1=1 or password='
            </fn:password>
        </fn:PerformFunction>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 1. **A sample SQL Injection Attack[4].**

4. DOS Attack Prevention Using WSDL Description

Denial of Service (DoS) attacks aim at reducing or completely eliminating a system's or service's availability. One can distinguish two kinds of DoS attacks: Protocol Deviation Attacks and Resource Exhaustion [5]. This section aims to build a validation scheme using WSDL definition of the web service to prevent most of XML DoS attacks. In this way, The SOAP message's structure belonging to a Web Service description is defined by informations spread all over the description document. The description must be traversed and the informations necessary for a specific service or operation must be merged into a message definition. The arrow in figure 2, shows how a Web Service description is traversed to determine the SOAP message's structure and to generate the appropriate XML Schema [6].

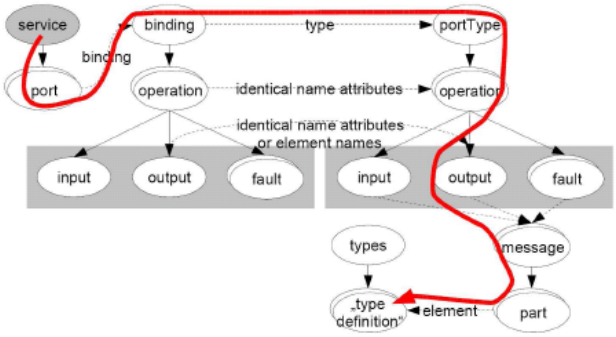

Figure 2. **Traversing WSDL to build the XML Scheme.[6]**

WSDL helps make a simple Scheme for XML requests. Such requests should be validated upon the scheme to pass the IDS. Such schemes mainly check the input parameters and avoid the requests containing deep loop-nested elements and XML DoS attacks.

5. Anomaly Detection Using XML Similarity

To improve the accuracy of the system, a normal dataset of requests are employed. The input request is compared against normal dataset of the requests. Such comparisions are made both on the structur and semantic contests based on the normal behaviors of the previous traces. Optimized Latent Semantic Analysis (LSA) is employed [7] to compare the input requests upon the normal dataset of requests.

6. XML Similarity Measurement

The XML document similarity between two documents d_x, d_y is computed as follow [8].

$$docSim(d_x, d_y) = \\ (contSim(d_x, d_y) * \lambda) + (structSim(d_x, d_y) * (1 - \lambda))$$

The document similarity is a combination of the content similarity and the structure similarity parameter. The λ ranging from 0 to 1, defines the importance of each similarity portition.

7. Structure Similarity Measure

The structure of an XML document relates to how the content in the XML document is structured. Information such as element names, data types, constraints, parents, ancestors, children, etc. can be used to discover the structural similarity among XML documents. To simplify the structure matching process, only, the most important property of the elements, are used for structure matching. The structure of an XML document is represented as a tree-based in which it is broken down into a collection of distinct paths. These paths are used to measure the structural distance between XML documents. Figure 3, shows a dataset of four XML documents [8].

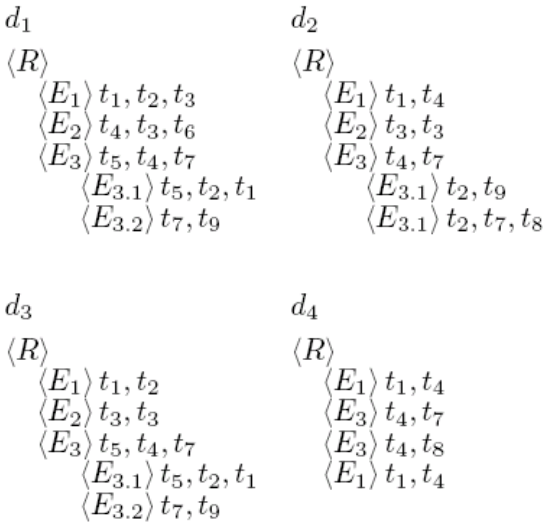

Figure 3. **Dataset D containg 4 XML documents[8].**

TABLE I. **SIMILARITY MATRIX OF DATASET STRUCTURE**

$Path/Doc$	d_1	d_2	d_3	d_4
R/E_1	1	1	1	2
R/E_2	1	1	1	0
$R/E_3/E_{3.1}$	1	2	1	0
$R/E_3/E_{3.2}$	1	0	1	0
R/E_3	1	1	1	2

To compute the structural similarity between two documents, below formula is used [8].

$$structSim(d_x, d_y) = \sqrt{\sum_{i=1}^{f} (p_{x,i} - p_{y,i})^2}$$

$p_{x,i}$ is the frequency of distinct path i in document x as shown in the above table.

8. Content Similarity

To determine similarity between XML documents, an optimized LSA algorithm is employed. Considering the documents in Figure 3. A set of distinct terms is extracted from the dataset D. A term-document matrix, $X_{m,n}$, where m is the number of terms and n is the number of documents in dataset D, is constructed as shown in Table 2 [8].

TABLE II. **TERM-DOCUMENT MATRIX**

Term/Do	d_1	d_2	d_3	d_4
t_1	2	1	2	2
t_2	2	2	2	0
t_3	2	2	2	0
t_4	2	2	1	4
t_5	2	0	2	0
t_6	1	0	0	0
t_7	2	2	2	1
t_8	0	1	0	1
t_9	1	1	1	0

The singular value decomposition (SVD) decomposes the term-document matrix, into three matrices , where U and V are left and right singular vectors respectively and S is a diagonal matrix of singular values ordered in decreasing magnitude.

$$X = USV^T$$

SVD can optimally approximate matrix X with a smaller sample of matrices by selecting k largest singular values and setting the rest of the values to zero. Matrix U_k of size $m*k$ consists of matrix V_k of size $n*k$ along with $k*k$ singular value matrix S_k .

$$\hat{X}_{m,n} = U_k S_k V_k^T$$

Matrix $\hat{X}_{m,n}$ is known to be the matrix of rank k which is closest in the least squares sense to X . Matrix U_k becomes the latent semantic kernel matrix. To compute the content similarity of two given XML documents, we use below formula.

$$contSim(d_x, d_y) = \frac{d_x^T P P^T d_y}{|P^T d_x||P^T d_y|}$$

Where matrix P is matrix U_k, and P is used as a mapping function to transform the two documents, d_x, d_y into concept space to determine the semantic association of document contents [8].

9. TF-IDF Information Retrieval

One of key issues in content similarity, is to properly score the best words for classification algorithms [9].In this algorithm , we use $tf * idf$ to update our kernel resulted from previous section. In this algorithm, the term specific weights in the document vectors are products of local and global parameters. The model is known as term frequency-inverse document frequency model. The weight vector for document d is $V_d = [w_{1,d}, w_{2,d}, ..., w_{N,d}]^T$,where $w_{t,d} = tf_{t,d} . \log \frac{|D|}{|\{d \in D | t \in d\}|}$ and $tf_{t,d}$ is term frequency of term t in document d .

$\log \frac{|D|}{|\{d \in D | t \in d\}|}$ is inverse document frequency , $|D|$ is the total number of documents in the document set and $|\{d \in D | t \in d\}|$ is the number of documents containing the term t.

Using the cosine the similarity between document d_j and d_l can be calculated as follow:

$$sim(d_i, d_l) = \frac{d_i . d_l}{\|d_i\| \|d_l\|} = \frac{\sum_{i=1}^{N} w_{i,j} * w_{i,l}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^{\ 2}} * \sqrt{\sum_{i=1}^{N} w_{i,l}^{\ 2}}}$$

10 . Optimized LSA

This section presents the optimization of our LSA algorithm. Such method have overhead to the enitire classifer due to the construction of the LSA matrixes. This is datasets. The computation time for building SVD of a matrix $m * n$ *is* $O(mn^2 + m^2 n + n^3)$ *[10].*

Hofmann [13] recommended the probabilistic latent semantic indexing (pLSI) which is utilised for written material modeling. In pLSI, a generative type for the documents and their remark occurrences is ushered in as follows. For a document-word pair (d,w). In pLSI, the latent variable z interpreted as the topics for words identifies the connection prospect of a bestowed document-word pair. Let us presume the number of written material N and remarks M are both fastened, hence z have a Multinomial dissemination with dimension N ×M. When we fix this latent variable to be K-dimensional disseminated, the latent variable truly acknowledges a clustering process in the connection space of written material and words. In [11] Hofmann z correspond to projection space for words.

Bose [13] introduced a novel process to diminish the complexity of makeup document/term matrix by adhesive binding and merging the documents. In this way, there is no loss for any terms since all the term are used for the kernel matrix construction .There are couple of ways to decide the document to fit in the bins. Random selection process decides the document steadily and left them illogically in the bins. While, support based approach decide the bulk noteworthy records steadily for the bins. It escapes the bias caused by illogical selection of the documents. Consider, the training dataset contains $D = \{D_1, D_2, D_3, \ldots, D_m\}$ *and document* D_i *has the maximum of* n *unique terms* $(t_{i1}, t_{i2}, \ldots, t_{in})$ *. After stemming and removing stop-words the frequencies are as follow* $(f_{i1}, f_{i2}, \ldots, f_{in})$*. The total frequency of term* T_j *is as follow.* $F_j = \sum_{i=1}^{m} f_{i,j}$ *. The importance of each document is computed. This process leads smaller amount of documents to fit in every bin. This greatly decreases the term-document matrix size. This algorithm dispenses the documnets steadily according to their importance and merge all the records in each bin. Previously, to compute the document importance, couple of factors are estimated, Support of term and the weight. Support of term defines the relative importance of the term in the whole corpus. Weight defines the importance of the term in the document. In this manner, ther is no bias toward the documents with frequent terms [14].*

The weigh of term t_{ij} is computed as follow.

$$W_j = \frac{f_{i,j}}{\sum_{j=1}^{n} f_{i,j}}$$

The size of dataset F , is total frequencies of terms in dataset:

$$F = \sum_{j=1}^{n} \sum_{i=1}^{m} f_{i,j}$$

Consider, $S = \{S_1, S_2, \ldots, S_n\}$ as the support of each term in corpus. The support shows the relative importance of each document among total corpus. It is computed as follow [12].

$$S_j = \frac{F_j}{F} = \frac{\sum_{i=1}^{m} f_{i,j}}{\sum_{j=1}^{n}\sum_{i=1}^{m} f_{i,j}}$$

Using support and weight of each term the importance of each documented is computed as follow.

$$DI_i = \sum_{j=1}^{n} W_j * S_j$$

If B be is the accumulation of q bins, $B = \{B_1, B_2, \ldots, B_q\}$, contains the same number of written documents from dataset in the bins. If BD is a accumulation of q numbers of documents $BD = \{BD_1, BD_2, \ldots, BD_q\}$, where BD_i is the joining documents after the dissemination algorithms in bin_i. After augmenting the frequency of merged terms in documents ,term with small number of frequencies (less than 5) or very high figures (above 5000) are considered as outliers. Such redundant terms are eliminated from term document matrix .In random selection, the algorithm selects equal number of documents for each bin at random. While for support based, the equal numbers of documents are selected according to their r importance. In this way, there are similar important documents in each bin. Such method avoids any bias caused by random selection and improves the accuracy of kernel. Bose proposed below algorithm for support-based selection of documents. Rederas are referred to [12] for more explanations.

It is inferred that pLSI is not a well-defined type, since it delights each document as an index and hence is not generalizable to new documents. Another obstacle of pLSI is that longer documents get higher weights in the type, which in addition suggests that the documents are not alone sampled. Moreover, pLSI uses a probabilistic method such as Naïve bayes algorithm to compute the probability of each topic. This implementation is obviously less efficient comparing to method proposed by Bose.

11. Serial Multiple Threshold Approach

It is obvious, attack detection is more sophisticated to discover with just one step. Therefeore, several steps are employed to define a request as attack. Each request is passed from multiple steps to assure the security of the request. In first step the request pre processed with malicious and unfamiliar codes and characters. Furthremore, as described in section 4 , the request is validated according to a normal XML scheme was explanide in section 4 [6]. There are many methods on combining different systems based on threshold criteria. This study employs serial sequence thresholding approch. The advantage is robust usage and keeping the accuracy and false negative ratio of the system [13].

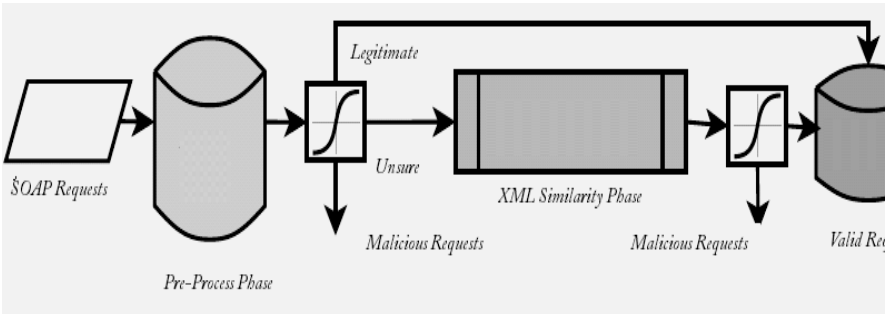

Figure 4. **A serial multiple threshod system**

Figure 4 shows a serial multiple threshold system. During the malicious request filtering process, the pre-process evaluation is performed. Two thresholds $T_l \succ T_s$ on this score are defined. Requests with legitimate score above T_l are accepted, skipping the content-based filter. Requests with a score lower than T_s are considered malicious. Requests with a score between the two thresholds, $T_l \succ f_i \succ T_s$, are passed to the XML-Similarity analyzer which defines the final status of the request. Similarity of each request is obtained from the maximum similarity of that request with the dataset. This score is compared with lower and higher bounds.

12. Results and Implementations

This section evaluates the proposed algorithm. WSDigger was employed [14] to simulate various types of SOAP attacks. A sample web service with 5 inputs was constructed. Moreover, a dataset of normal requests including 300 requests were generated. 50 malicious requests including XML and SQL injections wrere generated.

Experimet I was conducted to evaluate the overall efficiency of the proposed approach.Morever, precision and recall measurement metrics were employed [15].

TABLE III. **EVALUATING PARAMETERS**

	Assigned to C_i	**Not assigned to C_i**
Belonging to C_i	*tp*	*fn*
Not Belonging to C_i	*fp*	*tn*

C_i represents the clusters. The Precision shows the accuracy of the algorithm while the recall represents the integrity of suggestion algorithm.

$$\Pr ecision = \frac{tp}{tp + fp}$$

$$\Re call = \frac{tp}{tp + fn}$$

There maybe instances which the classifier does not categorize them. Therefore, it reduces

the Recall. We also use another parameter $F-Score$ can be computed as follow.

$$F = \frac{2 * \Pr ecision * \operatorname{Re} call}{\Pr ecision + \operatorname{Re} call}$$

Table 4, shows the results of each step. This study set 20 percent for the lower limit and 90 percent for the higher threshold. A rather high and low level thresholding was employed to let the system evaluate each step. Security manger can take a rigorous approach to limit such thresholds. Such limitation results in harder evaluation without losing false negative rate.

TABLE IV. **RESULTS OF INTRUSION DETECTION SYSTEM**

Using Pre-Process Evaluation	Using XML Scheme Validation	Using XML Similarity	Using Optimized XML-LSA	Using tf*idf	F (%)
False	eslaF	True	False	False	54.70 %
True	eurT	True	False	False	61.82 %
True	eurT	True	True	False	63.42 %
True	eurT	True	True	True	66.23 %

As inferred, simply using XML similarity measurement for defining SOAP requests had the least accuracy. While using pre-process evaluation and defining malicious code improved the accuracy. Moreover, using optimized LSA resulted in more accuracy and efficacy in the system. The last effort to improve the accuracy is to use $tf * idf$, was described in section 8. This process ranked all the unique terms in the corpus and improved the accuracy reasonably. Therefore, the promising way, is to use $tf * idf$ and optimized LSA along pre-process method and XML scheme evaluation to estimate each request based on the semantic and the structure.

Experiment II was conducted to change the lower limits and higher limits for the system. Using all steps, Table V, shows the results.

TABLE V. **RESULTS OF INTRUSION DETECTION SYSTEM**

	20-90 %	30-80%	40-70%
F (%)	66.23 %	65.46 %	62.72%

As seen, with restricting the limits the accuracy of the system is decreased.

In experiment III, TCPDump was employed [16] to distinguish the SOPA attacks. This is because of the fact that SOAP requests transmitted over TCP/IP protocol. Since, TCP packets were traced, there could not have enough information from a SOPA request. Therefore, XML Schemes or XML Structural similarity method on the segments was not implemented. To distinguish if a SOAP request is a malicious one, proposed algorithm was applied on the received packets.

For an incoming request, a multi-set, S_i of all encoded sequences was generated. This experiment considers normal packets as, N_d . Such packets were classified normal based on previous normal database of the requests. Anomaly is shown as the cardinality of multi-set

$|S_i - N_d|$. The incoming sample was classified as intrusion if anomaly counts exceeded a threshold t. This threshold was set low to avoid false positive detection rate.

$$t_i = \frac{|S_i - N_d|}{S_i}, \quad anomaly \quad if \quad t_i > t$$

Considering the threshold as little below 20 percent, the desired accuracies were achieved.

TABLE VI. **RESULTS OF TCPDUMP SEQUENCES**

Using Pre-Process Evaluation	Using XML Scheme Validation	Using XML Content Similarity	Using Optimized XML-LSA	Using tf*idf	F (%)
False	eslaF	True	False	False	46.12 %
True	eslaF	True	True	False	49..37 %
True	eslaF	True	True	True	53.41 %

13. Conclusions

This study introduced novel method to determine SOAP malicious requests from stream of input requests. First, the attacks containing malicious and unknown characters were determined. Secondly, the request was compared with and XML scheme based on a dataset of normal requests. Third, the request based on content and structural similarity was examined. This study employed LSA along with $tf * idf$ for semantic similruty. This approch was concluded as the promising way and it achived the highest accuracy comparing with other similar methods and combinational algorithms.

References

[1] Chan Gaik Yee, Wong Hui Shin, G.S.V.R.K. Rao, "An Adaptive Intrusion Detection and Prevention (ID/IP) Framework for Web Services," iccit, pp.528-534, 2007 International Conference on Convergence Information Technology (ICCIT 2007), 2007

[2] A. Stamos and S. Stender, “Attacking Web Services:The Next Generation of Vulnerable Enterprise Apps”, BlackHat2005, USA, 2005.

[3] K. Spett, “Blind SQL Injection:Are Your Web Applications Vulnerable?”. SPI Dynamics, 2005.

[4] Nirmal Dagdee, Urjita Thakar, "Intrusion Attack Pattern Analysis and Signature Extraction for Web Services Using Honeypots," icetet, pp.1232-1237, 2008 First International Conference on Emerging Trends in Engineering and Technology, 2008

[5] G¨unter Sch¨afer. Sabotageangriffe auf Kommunikationsstrukturen: Angriffstechniken und Abwehrmaßnahmen. PIK 28, pages 130–139, 2005.

[6] Gruschka, N. and Luttenberger, N.: "Protecting Web Services From DoS Attacks by SOAP Message Validation", IFIP International Federation for Information Processing, Springer Boston, Volume 201 (2006) 171-182

[7] Thomas Landauer, Peter W. Foltz, & Darrell Laham (1998). "Introduction to Latent Semantic Analysis" (PDF). Discourse Processes 25: 259–284.

[8] Tran, Tien and Nayak, Richi and Bruza, Peter D. (2008) Combining structure and content similarities for XML document clustering. In: 7th Australasian Data Mining Conference, 27-28 November 2008, Glenelg, South Australia.

[9] G. Salton, A. Wong, and C. S. Yang (1975), "A Vector Space Model for Automatic Indexing,"

Communications of the ACM, vol. 18, nr. 11, pages 613–620.

[10] Widdows, D., Ferraro, K.: Semantic vectors: A scalable open source package and online technology management application. In: Proceedings of the sixth international conference on Language Resources and Evaluation (LREC 2008), Marrakesh, Morroco (2008)

[11] T. Hofmann. Probabilistic Latent Semantic Indexing. In Proceedings of the 22nd Annual ACM SIGIR Conference, pages 50–57, Berkeley, California, August 1999.

[12] Bose, Aishwarya "Effective web service discovery using a combination of a semantic model and a data mining technique"master thesis,Faculty of Information technology,Brisbane, Qeesland , Australia,2008.

[13] R. Segal, J. Crawford, J. Kephart, and B. Leiba, "Spamguru: An enterprise anti-spam filtering system," in First Conference on Email and Anti-Spam CEAS 2004, Jul. 2004. [Online]. Available: http://www.ceas.cc/papers-2004/126.pdf

[14] Foundstone WSDigger, 2008, http://www.foundstone.com/us/resources/proddesc/wsdigger.htm

[15] Cleverdon, C. W. and Mills, J. The testing of index language devices. Aslib Proceeding. 15, 4, 106–130, 1963.

[16] V. Jacobson, C. Lres, S. McCanne, et al., "Tcpdump," http://www. tcpdump.org