PeANFIS-FARM Framework in Defending against Web Service Attacks

Gaik-Yee Chan[a], Chien-Sing Lee[a,*]
[a]Faculty of Computing and Informatics
(FCI), Multimedia University
Cyberjaya, Malaysia.
[a]gychan@mmu.edu.my

Swee-Huay Heng[b]
[b]Faculty of Information Science and Technology
(FIST), Multimedia University
Melaka, Malaysia
[b]shheng@mmu.edu.my

***Abstract*— Internet-enabled Web Service (WS) applications, such as e-commerce, are facing eXtensible Markup Language (XML)-related security threats. However, network and host-based intrusion (ID) and prevention (IP) systems and Web Service Security (WSS) standards are inadequate in countering against these threats. This paper presents a framework to mitigate XML/SOAP attacks. Our framework comprises of two intelligent models: the policy-enhanced adaptive neuro-fuzzy inference system (PeANFIS) and fuzzy association rule mining (FARM) model. Performance evaluation of each model indicates detection rate of greater than 99% and false alarm rate of less than 1%. In this paper, we aim to help the security administrator to decide which model to implement depending on the context of the situation. We present rule-based cases as examples to guide design and implementation decisions. Our future work shall see the implementation of the PeANFIS-FARM framework on a wider scale and in cloud computing.**

Keywords*— *intrusion detection; intrusion prevention; fuzzy association rule mining; e-commerce; web services

I. INTRODUCTION

The Internet, with its ability to transport huge amount of information over the World Wide Web (WWW) instantaneously, has become the driving force for the increasing growth of e-commerce applications. Additionally, the eXtensible Markup Language (XML)-based Web Services (WS) technology, with their many attractive features, such as platform independence, interoperability, and ease of use, have provided open access that allows users to access data and information beyond time, space and user group boundaries. Subsequently, more and more software applications, especially e-commerce applications are built on the WS platform. Most recently, it is mentioned in [1] that WS technology represents one of the most important technologies for e-service or e-commerce for over 60% of the companies under surveyed by Mckinsey Quarterly.

However, due to the inherent technological nature of WS, it creates a great security problem and has given rise to attacks such as oversized payload attacks, recursive payload attacks and coercive payload attacks, which in turn lead to Denial of Service (DoS) attacks. Existing intrusion detection and prevention (ID/IP) systems are mainly host or network-based, which do not address WS attacks, especially attacks related to XML and Simple Object Access Protocol (SOAP), two of the most basic building blocks for WS technology.

We propose to directly address XML/SOAP attacks at the Application Layer or Layer-7. An ID/IP framework with hybridized AI techniques in the form of model built within the application is proposed to mitigate the attacks mentioned above. One of the models is the policy-enhanced adaptive neuro-fuzzy inference system (PeANFIS) and the other is the fuzzy association rule mining (FARM) model. The design, development and performance evaluation of these two models have been discussed in [2, 3] respectively. In this paper, we aim to help the security administrator to decide which model to implement depending on the context of the situation.

II. WS ATTACKS, THREATS AND VULNERABILITIES

In summary, WS vulnerabilities and attacks can be categorized into 4 categories: a) misuse or abuse of UDDI/WSDL, such as parameter tampering; b) misuse or abuse of XML parser, such as CDATA, recursive payload and coercive parsing attacks; c) SOAP-related attacks such as SOAP oversized payload, SOAP Header attack, and replay attack; and d) XML content-related attacks such as XML content tampering, SQL injection and XML injection. The different types of attacks under each category and the relationship between the various types of attacks are shown in Fig. 1.

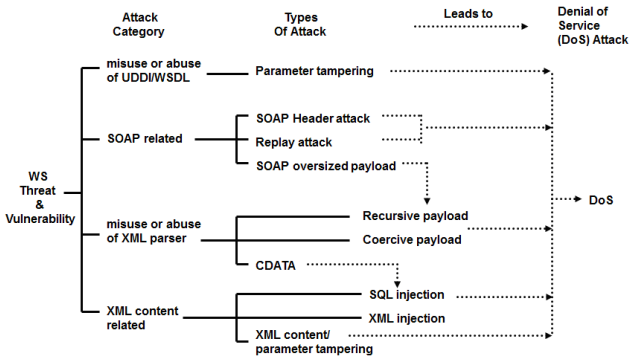

Fig. 1 WS attack categories and types of attacks

Referring to Fig. 1, all types of attacks under the 4 categories may eventually turn into Denial of Service (DoS) attacks. Additionally, these existing known attacks

*Corresponding author: Assoc. Prof. Dr. Chien-Sing Lee, csleectl@gmail.com, is currently affiliated with Faculty of Creative Industries, UTAR, Malaysia. She was formerly affiliated with FCI, MMU where the research works were carried out.

may evolve to new types of attacks as new hacking skills and tools evolve with time as well. According to [4], the WS-Security (WSS) Specification describes how the header part of the SOAP message is to be used for passing security information. An attacker using correct signature obtained illegitimately is able to bypass the XML authentication process. Furthermore, if the Web services are poorly implemented, the attacker can create complex SOAP header with source routing message to bypass security check. Consequently, the attacker can make use of this vulnerability to launch buffer overflow attack, SQL injection and XML injection attack to exploit the backend database or application. This may subsequently lead to DoS attack as seen in Fig. 1 or evolve to new type of attack.

Additionally, encrypted content can conceal attacks such as oversized payload, coercive parsing leading to XML injection or XML DoS [5]. As WSS does not define any direct countermeasures for DoS attacks, it is, therefore not 100% dependable. This cannot be tolerated as any WS's threats, vulnerability or attack can give rise to DoS attack as seen from Fig. 1.

WS attacks do exist and researches have been conducted to countermeasure against them. Some example of such researches can be found in [6,7,8,9,10,11]. In summary, these researches have shown that given good detection and prevention techniques, it is possible to develop an integrated ID/IP system to defend the WS applications against SOAP/XML attacks.

III. Fuzzy Logic and AI/Data Mining Techniques

To provide the best for both signature and anomaly-based intrusion detection, a hybrid approach combining Artificial Intelligence (AI) and data mining techniques has been actively applied in ID research. Some of these techniques are decision tree, rule-based technique, fuzzy logic with association rules and frequent episodes, genetic algorithm, neural network, Bayesian network, support vector machine and so on whose further details can be found in [12].

Many ID/IP researches have made use of fuzzy logic coupled with AI or data mining techniques in developing their ID/IP systems, models or frameworks for resolving the sharp boundary problem and feature selection problem hence, enhanced classification, improved detection and reduction of false alarm rates. Examples of these can be found in [13,14,15,16,17,18,19,20,21,22].

Through the above study, it is found that the ID systems which are built using fuzzy logic with AI/data mining techniques; ANFIS or FIS and fuzzy logic with association rule mining techniques are able to achieve high detection rates of 94% to 99% or greater and false alarm rates as low as 2% or lesser. However, these efficient and effective ID systems, whether signature-based and/or anomaly-based, are network-based ID systems for identifying abnormalities in network traffic for detection of Probe, DoS, User-to-Root (U2R) and Remote-to-Local (R2L), and not for the detection of WS application attacks related to SOAP and XML. Nevertheless, the above study has provided the feasibility and possibility that the proposed Application-based ID/IP system or model can also employ these techniques in defending against WS attacks related to SOAP and XML effectively.

We present our mitigation technique and process which incorporate hybridized AI techniques within an ID/IP framework for WS applications. It provides an added level of security protection for detection, prevention and prediction of XML/SOAP attacks. We elaborate on this ID/IP framework, PeANFIS-FARM, in subsequent section.

IV. The PeANFIS-FARM Framework

This PeANFIS-FARM framework is designed in such a way that intelligent intrusion detection, prevention and prediction techniques are embedded within the framework in the form of models. These fuzzy models can be instantiated to suit the different BI and network policies, thus contributing to greater sensitivity compared to its Boolean counterpart, and greater scalability and extensibility (due to the easy adoption and adaptation of fuzzy rules to different BI and network policies).

Additionally, input values, input size and SOAP size are validated by the models. The validated values are then matched with the 15 fuzzy rules obtained from the PeANFIS or the 20 fuzzy association rules obtained from the FARM model. In this way, any violation to normal profile is dynamically identified and immediate decision is taken to allow or deny access to the backend application or database. Based on the decision, further right action is taken to block, reject the request, terminate the subsequent activity or grant an alternative action. Refer to Fig. 2 for an overview of the PeANFIS-FARM framework.

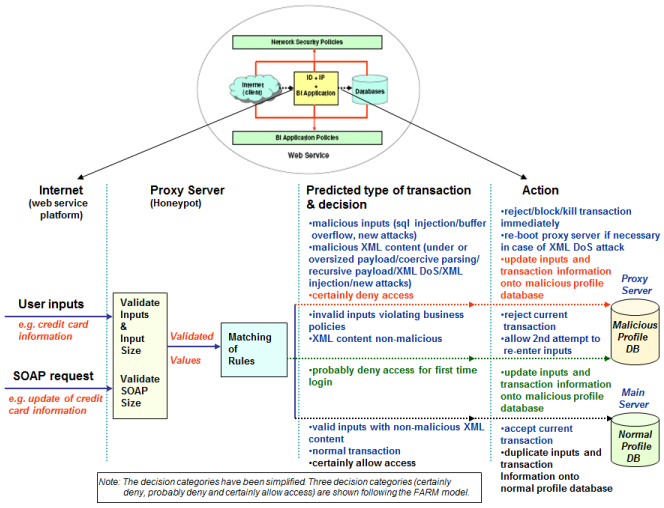

Fig. 2 An overview of PeANFIS-FARM framework

For the security administrator to choose which model to be integrated into the framework, we suggest to follow these criteria. Choose PeANFIS for ease of implementation and faster performance as it only needs to match with 15 fuzzy if-then rules obtained by validating 3 attributes, namely SOAP size, input size and decision. Otherwise choose FARM that requires matching of 20 fuzzy association rules obtained by validating 5 attributes, namely SOAP size, XML content, input values, input size and decision thus slowing down the performance of the framework. Another criterion is the number of decision categories. FARM has 3 decision categories (C Allow, C

Deny and P Deny) while PeANFIS has 4 (C Allow, C Deny, P Deny and P Allow). In real-life practical systems, it is still important to identify false positives and false negatives. Therefore, to enable FARM to detect false positives, then additional business policies have to be implemented to improve detection accuracy.

A. *The Predictive Model: PeANFIS*

The basic underlying architecture of an ANFIS consists of a fuzzy inference system (FIS) with adaptive neural networks (ANN). Although the FIS has no learning ability, but this deficiency is made up by the ANN which has powerful learning ability. To add to these basic abilities, for our predictive model, policy-enhanced ANFIS (PeANFIS), we impose business policies (Table I) onto the data set so as to limit the input values (credit card number, pin number, payment amount and email address) to lie within the valid range of values, input size and SOAP size to lie within the normal range of values. Outliers that contain invalid input values and out-of-range input or SOAP size can then be identified. These outliers representing anomalies or false alarms are then mitigated when the transactions are re-validated and decision re-classified. This in turn will strengthen the model to be more accurate as false alarms are reduced.

TABLE I. BUSINESS POLICIES

No.	Transaction inputs	Policy Description
1	User ID	Minimum 6, maximum 12 alpha-numeric with no special characters, mandatory input
2	Password	Minimum 6, maximum 12 alpha-numeric with no special characters, mandatory input
3	Credit Card #	16-digit numbers only, mandatory input
4	3-digit pin	3-digit numbers only, mandatory input
5	Payment amount	Float values with 2 decimals, 9999999.99, mandatory input
6	Email address	Minimum 15, maximum 35 alpha-numeric with no special characters, except @, mandatory input

In order to provide more meaningful means to perform the intrusion detection, prevention or decision functions, fuzzy logic is used to convert the numerical attributes to fuzzy attributes. For example, SOAP size is "greatly oversized" when it is greatly out-of-range of the upper-bound of 437 bytes and input size is "small" when it is way out of its lower-bound of 38 bytes. In this way, a set of meaningful linguistic labels represented by fuzzy sets on the domain of the quantitative attributes are mapped to a new domain. Refer to Table II for the fuzzy associative matrix between SOAP size, input size and decision.

TABLE II. FUZZY ASSOCIATIVE MATRIX-PEANFIS

SOAP / INPUT Size	Small	Normal	Large
Greatly Undersized	C Deny	P Deny	C Deny
Slightly Undersized	C Deny	P Allow	C Deny
Normal Range	C Deny	C Allow	C Deny
Slightly Oversized	C Deny	P Allow	C Deny
Greatly Oversized	C Deny	P Deny	C Deny

C Deny: Certainly Deny Access P Deny: Probably Deny Access
C Allow: Certainly Allow Access P Allow: Probably Allow Access

The quantitative values of SOAP size and input size are transformed into the range of values that fit into the PeANFIS model by the fuzzification and defuzzification process through the Memdani Fuzzy Inference System (FIS) built using MATLAB 5.3R. For further details of design, implementation and performance evaluation of this model, refer to our works at [2].

The PeANFIS also demonstrates the ability to detect and prevent known attacks such as SOAP oversized payload; detect, prevent or predict XML DoS caused by coercive parsing and recursive payload attacks with the possibility of discovering new attack; detect, prevent or predict XML parameter or content tampering attack with the possibility of discovering new attack. Refer to Table III for some examples of such cases.

TABLE III. PEANFIS CASES

Case	Input values	*Input/SOAP Size & Decision	Validated Inputs & XML Contents
1	Credit Card# : * 3-digit pin : * Amount : * Email address : AAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAA	Input size : 103 (Large) SOAP size : 476 (Normal) Decision: **C deny**	Input values: malicious XML content: non-malicious **Buffer Overflow detected & prevented**
2	Credit Card# : 123456789012345 3-digit pin : 111 Amount : 3.45 Email address : @@@@@@@@ @@@@@@@@@@@@@@@	Input size :498 (Large) SOAP size: 871 (Greatly Oversized) Decision: **C deny**	Input values: malicious (Buffer Overflow detected & prevented) **Predictive of coercive parsing/ recursive payload attacks**
3	Credit Card# : 1 3-digit pin : 1 Amount : 1 Email address : 1	Input size : 4 (Small) SOAP size: 377 (Slightly Undersized) Decision: **C deny**	Input values: malicious XML content: malicious **Malicious inputs detected & prevented. Predictive of XML content tampering attack.**
4	Credit Card# : 'or 1 = 1 or '' =' 3-digit pin : 'or 1 = 1 or '' =' Amount : 'or 1 = 1 or '' =' Email address : 'or 1 = 1 or '' =''	Input size : 72 (Large) SOAP size : 445 (Normal) Decision: **C deny**	Input values: malicious XML content: non-malicious **SQL injection attack detected & prevented**
5	Credit Card# : 2345 3-digit pin : 123 Amount : 345678 Email address : test@yahoo.com	Input size : 27 (Small) SOAP size: 439 (Normal) Decision: **C deny**	Input values: non-malicious XML content: hidden scripts **(...<foo><![CDATA['or 1=1 or''='']]></foo>)** **CDATA attack detected & prevented**

* The third column makes reference to Table II

B. *The Predictive Model: FARM*

Association rule mining is a data mining technique used to investigate the possibility of simultaneous occurrence of related item sets in a large database. Two parameters, *support* and *confidence* are used to describe the "interestingness" of a rule. All item sets that have support greater than or equal to the user specified minimum support are generated. The basic *Apriori* algorithm works in two steps. First, it finds all frequent item sets according to the minimum *support* value. Next, it generates all the association rules that have minimum *confidence* from the frequent item sets found in the first step, generally known as pruning. Pruning is based on the fact that if an item set is frequent all its subsets are frequent as well. Therefore, the algorithm discards every candidate item set that has an infrequent subset. Through this process of mining and pruning, a vast amount of rules can be generated.

However, not all discovered rules are *interesting* enough to be of good use for decision making. To find *interesting* or valid rules from our FARM, a series of sensitivity and extensibility analysis are conducted on various datasets using the open source software WEKA [23]. To add to this basic capability for our FARM, we

impose business policies (refer to Table I) onto the data set so as to limit the input values to lie within the valid range of values. Outliers that contain invalid input values and out-of-range input or SOAP size can then be identified.

Additionally, we use fuzzy logic to convert numerical attributes to fuzzy attributes through the fuzzification process. For example, input size is "greatly oversized" when it is greatly out-of-range of the upper limit of 64 bytes, or SOAP message size is "extremely oversized", when it is extremely-out of range by many folds above the upper limit of 437 bytes. Referring to Table IV, it is observed that there is a correlation between SOAP size and XML content.

TABLE IV. FUZZY ASSOCIATION MARTIX-FARM

Attributes	Values		
SOAP size	Matched		
XML content	Non-malicious		
Input values	Valid	Malicious/New	Invalid
Input size	Normal	Normal/ Out-of-range/ Extremely-out	Normal/ Out-of-range/ Extremely-out
Decision	C Allow	C Deny	P Deny
SOAP size	Not-matched		
XML content	Malicious		
Input values	Valid	Malicious/New	Invalid
Input size	Normal	Normal/ Out-of-range	Normal/ Out-of-range
Decision	C Deny	C Deny	C Deny
SOAP size	Extremely-not-matched		
XML content	New		
Input values	Valid	Malicious/New	Invalid
Input size	Normal	Extremely-out	Extremely-out
Decision	C Deny	C Deny	C Deny

Consequently, our predictive model FARM is able to detect and prevent known attacks as well as predict or discover new or unknown attacks as seen from the following five cases of certainly deny access (Table V), indicating either inputs or XML content is malicious. Detect and prevent known attacks such as SQL injection (Table V: Case 1), buffer overflow (Table V: Case 2), SOAP oversized payloads, cross site scripting-XSS attacks (Table V: Case 3). Detect and prevent XML DoS caused by coercive parsing and recursive payload attacks with the possibility of discovering new attack (Table V: Case 4) and detect and prevent XML parameter or content tampering attack with the possibility of discovering new attack (Table V: Case 5).

Subsequently, by segregating the anomalies from the normal using our FARM has enabled us to determine frequently occurring features from the set of fuzzy association rules obtained. This in turn helps the security administrator to prioritize which feature to focus on in the future thus addressing the features selection problem. This paper presents only a summarized view of the FARM model. Further discussion regarding design, implementation and performance evaluation of the model can be seen in our works at [3].

TABLE V. FARM CASES

Case	Input values	Size & decision	Comments
1	Credit Card#: 'or 1 = 1 or '' =' 3-digit pin : 'or 1 = 1 or ' '=' Amount : 'or 1 = 1 or '' =' Email : 'or 1 = 1 or '' =' Refer to Table I Inputs violate business policy#: 3,4,5,6	Refer to Table IV Input size : 72 (Out-of-range) SOAP size: 437 (64 + 373 = 437) (Matched) Decision :**C deny**	Validate inputs: Inputs contain malicious SQL codes. Input size is out-of-range. SOAP size matched, XML content is non-malicious. Decision is to certainly deny access. **SQL injection** attack detected & prevented.
2	Credit Card#: * 3-digit pin : * Amount : * Email : ********** ********** Refer to Table I Inputs violate business policy#: 3,4,5,6.	Refer to Table IV Input size : 103 (Extremely-out) SOAP size: 476 (103+373 = 476) (Matched) Decision :**C deny**	Validate inputs: Inputs contain malicious codes. Input size is extremely-out. SOAP size matched, XML content is non-malicious. Decision is to certainly deny access. **Buffer overflow** detected & prevented.
3	Credit Card#: 1234567890123456 3-digit pin : 555 Amount : 3.33 Email : test1@yahoo.com Refer to Table I Inputs do not violate business policy#:3,4,5, 6.	Refer to Table IV Input size : 38 (Normal range) SOAP size: 439 (38 + 373 = 411) **SOAP oversized** (439 – 411 = 28) Decision :**C deny**	Validate inputs: Valid inputs with no violation of business policies. Input size is in normal range. SOAP size not-matched, XML content is malicious with hidden scripts: **<script>alert("hi")</script>** Decision is to certainly deny access. **XSS** attack detected & prevented.
4	Credit Card#: 1234567890123456 3-digit pin : 555 Amount : 3.33 Email :test1@yahoo.com Refer to Table I Inputs do not violate business policy#: 3,4,5,6.	Refer to Table IV Input size : 38 (Normal range) SOAP size: 1157 (38 + 373 = 411) **SOAP oversized** by many folds Decision :**C deny**	Validate inputs: Valid inputs with no violation of business policies. Input size is in normal range. SOAP size is extremely-not -matched. XML content is malicious, predictive of **XML DoS or new attack**. Decision is to certainly deny access. **SOAP oversized payload** attack detected & prevented.
5	Credit Card#: 1234567890123456 3-digit pin : 555 Amount :102.35 Email :fayfay@yahoo.com Refer to Table I Inputs do not violate business policy#: 3,4,5,6.	Refer to Table IV Input size : 41 (Normal range) SOAP size: 300 (38 + 373 = 411) **SOAP undersized** greatly Decision :**C deny**	Validate inputs: Valid inputs with no violation of business policies. Input size is in normal range. SOAP size is extremely-not- matched. XML content is malicious, predictive of **XML content tampering or new attack**. Decision is to certainly deny access. **XML content tampering** attack detected & prevented.

C. *The Mitigation Process*

In summary, the steps involved in the mitigation process for the Web service request transaction are presented in Fig. 3, based on the assumption that the User ID and password are validated to be both valid.

1. Capture transaction inputs (INPUTS: credit card number, pin, email address, amount payable, and SOAP size).
2. Compute service request's input size and validate INPUTS using business policies (Table I).
3. Categorize validated INPUTS as valid, invalid or malicious, input size to be in normal range or out-of-range and SOAP size to be in normal range, undersized or oversized.
4. For PeANFIS, go to step 4 (a) else for FARM, go to step 4 (b).
   (a) Match the validated INPUTS with the fuzzy rules derived from the PeANFIS (Table II); go to step 5.
   (b) Match the validated INPUTS with the fuzzy association rules obtained from the FARM model (Table IV); go to step 6.
5. Case i: If input size and SOAP size are in the normal range, then certainly allow access and accept the transaction.
   Duplicate INPUTS and transaction information onto the normal profile database of the main server.
   End of transaction, go to step 7.
   Case ii: If input size is in the normal range but SOAP size is slightly under or oversized, then probably allow access and accept the transaction temporarily.
   Duplicate INPUTS and transaction information onto the malicious profile database of the proxy server for further analysis and confirmation of non-intrusion or anomaly.
   Duplicate INPUTS and transaction information onto the normal profile database of the main server if it is confirmed to be a genuine valid transaction.
   Repeat step 4(a) if otherwise.
   End of transaction, go to step 7.
   Case iii: If input size is in normal range but SOAP size is greatly over or undersized, then probably deny access and allow another attempt to re-enter INPUTS.
   Duplicate INPUTS and transaction information onto the malicious profile database of the proxy server.
   Repeat steps 1-4.
   Case iv: If input size is out-of-range regardless of SOAP size, then certainly deny access and reject the transaction.
   Duplicate INPUTS and transaction information onto the malicious profile database of the proxy server.
   End of transaction, go to step 7.
6. Case i: If INPUTS are valid with input size and SOAP size in normal range and XML content is non-malicious, then allow access and accept the transaction.
   Duplicate INPUTS and transaction information onto the normal profile database of the main server.
   End of transaction, go to step 7.
   Case ii: If INPUTS are invalid in violation of business policies with input size and SOAP size in normal range and XML content is non-malicious, then probably deny access but allow another attempt to re-enter INPUTS.
   Duplicate INPUTS and transaction information onto the malicious profile database of the proxy server.
   Repeat steps 1-4.
   Case iii: If INPUTS (input values or XML content) contain malicious or unknown values, then deny access and reject the transaction.
   Duplicate INPUTS and transaction information onto the malicious profile database of the proxy server.
   End of transaction, go to step 7.
7. End of mitigation process.

Fig. 3 The mitigation process

V. CONCLUSION AND FUTURE WORK

SOAP and XML-related attacks do exist at the Application layer and can be detected and prevented by validating input values, input size and SOAP size. We have applied fuzzy logic in our models to define a set of meaningful linguistic labels represented by fuzzy sets on

the domain of the quantitative attributes and map to a new domain. Further analyses of the datasets allow us to discover patterns among the attributes. Subsequently, by restricting the inputs using business policies, we have further strengthened the PeANFIS-FARM framework to be able to detect, prevent and predict XML/SOAP attacks, either with the fuzzy rules of the PeANFIS or the fuzzy association rules of the FARM. Hence, this novel ID/IP framework significantly provides a viable added layer of security protection for WS applications.

Our future work shall see the implementation of the PeANFIS-FARM framework in cloud computing. The implementation shall make use of real-world WS application to capture the normal and attack data for optimum evaluation besides detection and false alarm rates for effectiveness, and on 'time' performance for efficiency.

REFERENCES

[1] M. Sellami, O. Bouchaala, O, W. Gaaloul, and S. Tata, "Communities of web service registries: construction and management," The Journal of Systems and Software 2013, vol. 86, pp. 835– 853.

[2] G.Y. Chan, C.S. Lee, and S.H. Heng, "Policy-enhanced ANFIS model to counter SOAP related attacks," Knowledge-Based Systems 2012, vol. 35, pp. 64-76.

[3] G.Y. Chan, C.S. Lee, and S.H. Heng,"Discovering fuzzy association rule patterns and increasing sensitivity analysis of XML-related attacks," Journal of Network and Computer Applications 2013, vol. 36(2), pp. 829-842.

[4] J. Holgersson and E.Soderstrom,"Web service security-vulnerabilities and threats within the context of WS-security," Proceedings of IEEE 4th Conference on Standardization and Innovation in Information Technology 2005, pp. 138-146.

[5] M. Jensen, N. Gruschka, and R. Herkenhoner, "A survey of attacks on web services," Journal of Computer Science-Research and Development 2009, vol. 24(4), pp. 185-197.

[6] P. Lindstrom, "Attacking and defending web services," A Spire Research Report, 2004. Available: http://www.forumsys.com/resourc es/resources/whitepapers/Attacking_and_Defending_WS.pdf

[7] Y.S. Loh, W.C. Yau, C.T. Wong, and W.C. Ho, "Design and implementation of an XML firewall," Proceedings of the 2006 International Conference on Computational Intelligence and Security pp. 1147-1150.

[8] X. Ye, "Countering DDoS and XDoS attacks against web services," Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008, pp. 346-352.

[9] V. Relan and B. Sonawane,"Detection and mitigation of web services attacks using markov model, 2009, CMSC 678: Machine Learning Project.Available:http://userpages.umbc.edu/~relan1/CMSC%2067 8%20Project%20Report.pdf

[10] U. Thakar, N. Dagdee, and S. Varma, "Pattern analysis and signature extraction for intrusion attacks on web services," International Journal of Network Security & Its Applications, 2010, vol. 2(3), pp. 190-205.

[11] C.I. Pinzón, J. Bajo, J.F. De Paz, and J.M. Corchado,"S-MAS: an adaptive hierarchical distributed multi-agent architecture for blocking malicious SOAP messages within web services environments," Expert Systems with Applications 2011, vol. 38, pp. 5486–5499.

[12] G. Kumar, K. Kumar, and M. Sachdeva, "The use of artificial intelligence based techniques for intrusion detection: A review," Artificial Intelligence Review 2010, vol. 34(4), pp. 369-387.

[13] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, "Intrusion detection using fuzzy association rules," Applied Soft Computing 2009, vol. 9, pp. 462–469.

[14] K.X. Wu, J. Hao, and C.H. Wang, "Intrusion detection based on fuzzy association rules," Proceedings of the 2010 International Symposium on Intelligence Information Processing and Trusted Computing, pp. 200-203.

[15] M.M.T. Jawhar and M. Mehrotra,"Design network intrusion detection system using hybrid fuzzy-neural network," International Journal of Computer Science and Security 2010, vol. 4(3), pp. 285-294.

[16] A. Zainal, M.A. Maarof, and S.M. Shamsuddin, "Ensemble classifiers for network intrusion detection system," Journal of Information Assurance and Security 2009, vol. 4, pp. 217-225.

[17] X.D. Hoang, J. Hu, and P. Bertok, "A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference," Journal of Network and Computer Applications 2009, vol. 32(6), pp. 1219-1228.

[18] T. Subbulakshmi, S.M. Shalinie, C.S. Reddy, and A. Ramamoorthi, "Detection and classification of DDoS attacks using fuzzy inference system," Recent Trends in Network Security and Applications, Communications in Computer and Information Science 2010, vol. 89(1), pp. 242-252.

[19] M. Sheikhan and Z. Jadidi,"Misuse detection using hybrid of association rule mining and connectionist modeling," World Applied Sciences Journal 2009, vol. 7, Special Issue of Computer & IT, pp. 31-37.

[20] M. Sheikhan and M.S. Rad, "Misuse detection based on feature selection by fuzzy association rule mining," World Applied Sciences Journal 2010, vol. 10, Special Issue of Computer & Electrical Engineering, pp. 32-40.

[21] S. Mabu, C. Chen, N. Lu, K. Shimada, and K. Hirasawa, "An Intrusion-detection Model Based on Fuzzy Class-association-rule Mining Using Genetic Network Programming," 2011 IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 41(1), pp. 130-139.

[22] V. Markam and L.S.M. Dubey, "A general study of associations rule mining in intrusion detection system," International Journal of Emerging Technology and Advanced Engineering 2012, vol. 2(1), pp. 347-356.

[23] WEKA software (weka-3-6-4jre.exe): Available http://www.cs.waikato.ac.nz/ml/weka/