PeANFIS-FARM Framework in Defending against Web Service Attacks

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

Swee-Huay Heng\textsuperscript{b}
\textsuperscript{b}Faculty of Information Science and Technology (FIST), Multimedia University
Melaka, Malaysia
\textsuperscript{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](image)

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, cslleecl@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.

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

<table>
<thead>
<tr>
<th>No.</th>
<th>Transaction inputs</th>
<th>Policy Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User ID</td>
<td>Minimum 6, maximum 12 alpha numeric with no special characters, mandatory input</td>
</tr>
<tr>
<td>2</td>
<td>Password</td>
<td>Minimum 6, maximum 12 alpha numeric with no special characters, mandatory input</td>
</tr>
<tr>
<td>3</td>
<td>Credit Card</td>
<td>12-digit number, mandatory input</td>
</tr>
<tr>
<td>4</td>
<td>3-digit pin</td>
<td>3-digit numeric, mandatory input</td>
</tr>
<tr>
<td>5</td>
<td>Payment amount</td>
<td>All values will be 2 decimals. 00000.00 mandatory input</td>
</tr>
<tr>
<td>6</td>
<td>Email address</td>
<td>Minimum 15, maximum 35 alpha numeric with no special characters, except @, mandatory input</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>SOAP / INPUT SIZE</th>
<th>Small</th>
<th>Normal</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greatly Under-sized</td>
<td>C Deny</td>
<td>P Deny</td>
<td>C Deny</td>
</tr>
<tr>
<td>Slightly Under-sized</td>
<td>C Deny</td>
<td>P Allow</td>
<td>C Deny</td>
</tr>
<tr>
<td>Normal Range</td>
<td>C Deny</td>
<td>C Allow</td>
<td>C Deny</td>
</tr>
<tr>
<td>Slightly Oversized</td>
<td>C Deny</td>
<td>P Allow</td>
<td>C Deny</td>
</tr>
<tr>
<td>Greatly Oversized</td>
<td>C Deny</td>
<td>P Deny</td>
<td>C Deny</td>
</tr>
</tbody>
</table>

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 Memdami 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

<table>
<thead>
<tr>
<th>Case</th>
<th>Input Values</th>
<th>Input/SOAP Size &amp; Decision</th>
<th>Validated Inputs &amp; XML Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Credit Card</td>
<td></td>
<td>Input size: 103 (Large) SOAP size: 476 (Normal) Decision: C deny</td>
</tr>
<tr>
<td>2</td>
<td>Credit Card</td>
<td></td>
<td>Input size: 498 (Large) SOAP size: 871 (Normal) Decision: C deny</td>
</tr>
<tr>
<td>3</td>
<td>Credit Card</td>
<td></td>
<td>Input size: 4 (Small) SOAP size: 377 (Small) Decision: C deny</td>
</tr>
<tr>
<td>4</td>
<td>Credit Card</td>
<td></td>
<td>Input size: 72 (Large) SOAP size: 465 (Large) Decision: C deny</td>
</tr>
<tr>
<td>5</td>
<td>Credit Card</td>
<td></td>
<td>Input size: 27 (Small) SOAP size: 439 (Small) Decision: C deny</td>
</tr>
</tbody>
</table>

* The third column makes reference to Table IV

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 “interesteningess” 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.

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 IV. FUZZY ASSOCIATION MATRIX-FARM

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Values</th>
<th>XML content</th>
<th>SOAP size</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input values</td>
<td>Valid</td>
<td>Malicious/Nov</td>
<td>Invalid</td>
<td></td>
</tr>
<tr>
<td>Input size</td>
<td>Normal</td>
<td>Normal/Out-of-range</td>
<td>Normal/Extremely-out</td>
<td></td>
</tr>
</tbody>
</table>

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).

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.

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


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