Journal of Computer Science 7 (2): 255-261, 2011
ISSN 1549-3636

Handling Web and Database Requests Using Fuzzy Rules for Anomaly Intrusion Detection

Selvamani Kadirvelu and Kannan Arputharaj
Department of Computer Science and Engineering
Anna University, Chennai, Tamilnadu, India

Abstract: Problem statement: It is necessary to propose suitable detection and prevention mechanisms to provide security for the information contents used by the web application. Many prevention mechanisms which are currently available are not able to classify anomalous, random and normal request. This leads to the problem of false positives which is classifying a normal request as anomalous and denying access to information. **Approach:** In this study, we propose an anomaly detection system which will act as a Web based anomaly detector called intelligent SQL Anomaly detector and it uses decision tree algorithm and a feedback mechanism for effective classification. **Results:** This newly proposed and implemented technique has higher probability for reducing false positives which are the drawbacks in the earlier systems. Hence, our system proves that it detects all anomalies and shows better results when compared with the existing system. **Conclusion:** A refreshing technique to improve the detection rate of web-based intrusion detection systems by serially framing a web request anomaly detector using fuzzy rules has been proposed and implemented and this system proves to be more efficient then the existing earlier system when compared with the obtained results.

Key words: SQL anomaly detection, fuzzy rule, access control, feedback propagation, Intrusion Detection System (IDS), anomaly detection system, propagation algorithm, encrypted data, HTTP proxy

INTRODUCTION

The Web applications provide easy access to the relevant and necessary information through web pages to the users. In many cases, web applications include client-side components such as JavaScript code that interact with server-side components. The server-side components often access the application's data stored in a back-end database. On the other hand, the application-specific code is often developed under strict time constraints by programmers with little experience and security training. As a result, vulnerable web-based applications are often exposed to the entire Internet, creating easily-exploitable entry points for the compromise of entire networks. Hence, these vulnerable web-based applications has attracted the attention of malicious hackers, who see in web-based applications relatively easy ways for exploitation and try to access sensitive information, which might lead to a monetary gain. Also the previous systems which use only signature based techniques are not proved to be efficient as they can detect only the attacks which are already available in signatures where in new type of attacks cannot be detected.

Therefore, the security of web applications should be handled by means of careful design and exhaustive security testing mechanisms to protect such web applications. Hence, the security conscious development methodologies should be incorporated by introducing an Intrusion Detection System which can detect and prevent any threats to the web application. Web application attacks can be detected effectively and possibly prevented by designing intelligent Intrusion Detection System (IDS) that use signature-based techniques along with intelligent rules for effective detection and prevention.

To detect attacks that are trim to specific web applications, an anomaly detection system has been proposed in this research work which provides security to web applications. This system uses C4.5 classification algorithm to classify the normal usage profiles associated with the applications and the anomalous behaviors. These classified results are then used as a basis to determine if a request is "anomalous" and to prevent such requests.

Corresponding Author: Selvamani Kadirvelu, Department of Computer Science and Engineering, Anna University, Chennai, Tamilnadu, India

This approach has shown considerable detection accuracy in comparison with the research prototypes and commercial products which were developed based on this idea. Ironically Anomaly detection systems have a tendency to include false positives and false negatives which are not considered by the earlier systems. Though False Positive is an event signaling IDS to produce an alarm when no attack has taken place and false negative is a failure of IDS to detect an actual attack are of different types of security concerns, both should be handled to provide a better security system.

To alleviate these problems, we propose a web based anomaly detection system called intelligent SQL anomaly detection system which uses C4.5 classification algorithm. This detection system improves the rate of true positives but also decreases false positives. In order to achieve this, we introduce an intelligent solution based on rules and feedback propagation. The idea is to duplicate a web site on two or more sib web servers with different levels of privilege that uses different types of keys. (i.e., different levels of access to database). Anomaly scores generated from this web based anomaly detection system are used by C4.5 the reverse HTTP proxy and feedback propagation algorithm.

In order to provide suitable access control mechanisms three user level privileges namely the super user (privileged user), application programmer and naïve user are proposed and implemented in this system. The first two levels are provided with secure private keys where as the third level user (naïve user) is provided with a group key.

Related works: The detection of web-based attacks has recently received significant attention among the researchers in the area of web security because of the improved necessity and use of web applications. For example, (Almgren *et al*., 2000) developed a system that analyzes web logs looking for patterns of known attacks.

A different analysis is executed in where the detection process is combined with the web server application itself. (Syurahbil *et al*., 2009) presented a novel method to find intrusion characteristics for IDS using decision tree machine learning. They used the method to generate rules for classification by ID3 algorithm of decision tree. Their system demonstrates that it is possible to achieve better detection results when taking advantage of the specificity of a particular application domain. All these techniques required prior signatures of attack types. Such anomaly intrusion detection systems can detect novel attacks. They are also capable of correctly classifying legitimate requests as malicious and normal. Thus, it is important to develop strategies that can classify the results correctly and can deal with false positives.

A different approach is presented for health care system using Intrusion detection system that uses Negative Selection Algorithm, (Sundararajan and Shanmugam, 2010) where an intrusion detection system is used to separate out normal events so that the negative selection detectors are capable differentiating the normal and abnormal behaviors and also reduces the false positive ratio that increases the system performance.

In the approaches described above, each individual action (e.g., an HTTP request) is be either allowed or denied. In contrast to that, we provide facilities to serve all requests which are legitimate and others are sent with lower level privileges for further investigation.

The hardened system presented a technique which is composed of an anomaly detection system that uses abstract payload execution and payload shifting techniques to identify web requests that might contain attacks that exploit memory (Toth and Kruegel, 2002) violations (e.g., buffer overflows and heap overflows). The requests that are identified as anomalous are then marked appropriately and processed by a hardened, "shadow" version of the web server. Even though this system provides facilities to all users, no feedback mechanism is provided to change the access control policies.

In this study, we propose an intelligent anomaly IDS for reducing false positive rates, while there are few similarities between the approach proposed (Mehdi *et al*., 2007) and ours, there are several major differences, the most significant of which is their system's limitation to a single class of attacks, namely that of control flow data corruption. Prior work by Lee *et al*. has considered the application of learning techniques to the problem of identifying web-based attacks on databases (Tombini, E., H. Debar, L. M´e and M. Ducasse, 2004). The structure matching approach proposed by Lee addresses this problem; but mimicry attack is possible against this detection mechanism, while our system is significantly more resistant to this class of attack because we use rules and feedback mechanism in addition to the classification using C4.5 algorithm. Comparing with all the works available in the literature, the work presented in this study is different in many ways. First, we use a web database which can store only encrypted data.

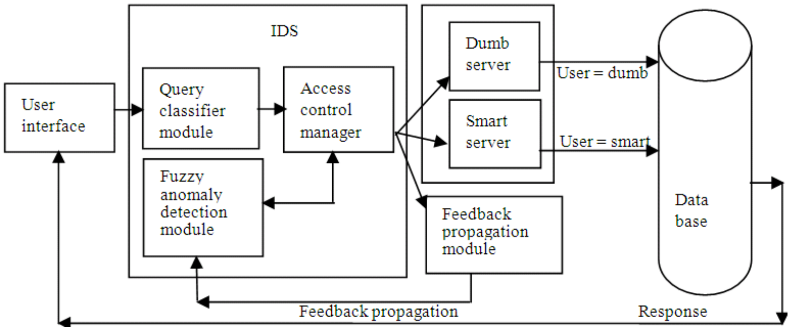

Fig. 1: Architecture of anomaly intrusion detection system

The legitimate users are provided with a private key with which they can decrypt the contents and then use them. Second, we serve all the requests from different users with different privilege levels. Third, we use intelligent rules for effective access control which permit users only after satisfying the required constraints. Finally, we focus an anomaly intrusion detection and prevention through classification and feedbacks thus reducing false positive and increasing true positive rates.

System Architecture: The goal of detecting unknown attacks against web applications as well as reducing false positives requires new techniques in IDS. In this study, we propose an architectural framework for intelligent anomaly intrusion detection system and intelligent techniques for access control as shown in Fig. 1. Hence, first we create a log file consisting of non anomalous records which are obtained from a classifier. The incoming requests are sent to the query classifier, which checks the incoming requests whether it is normal or anomalous by applying the query classification algorithm. If the request is normal, it is simply let to pass through else an alert is generated and the privilege level is reduced by one level and then it is allowed to access the database instead of completely denying the access.

This helps in reducing the false negatives. Then the request is made to pass through a query classifier which forwards it to appropriate server depending upon the anomaly score generated using the feedback propagation algorithm used in the web anomaly detection and prevention system. The system consists of two servers namely smart and dumb servers. Requests coming to the server do not get access to the database directly. They are made to pass through an intelligent SQL access control system which scrutinizes the SQL query that accesses the database and checks for anomaly score using rules.

Then this message is sent back to the web anomaly detector as a feedback and a profile is created for anomalous queries. The individual components of architecture are explained in more detail.

User interface: This is responsible for accepting user queries and to generate http requests. This module is also responsible for displaying the query results to the user after the query has been executed by the web server.

Query classifier: This access control manager utilizes anomaly scores sent by the anomaly detection module to detect attacks against back-end SQL databases by setting privilege levels. This module is deployed between web based applications and the back-end database server. The SQL queries sent by the applications are captured and sent to the IDS for analysis. The query classifier module parses each incoming SQL query and classifies them using C4.5 algorithm and then sent it to the fuzzy anomaly detection module which applies a fuzzy rules to generate score and level of anomaly for the query. These anomaly scores are sent to the access control manager which forwards the query to suitable web servers based on the anomaly scores.

Fuzzy anomaly detection module: The fuzzy decision manager first extracts from the requested URL, the path to the web application being invoked, along with the arguments passed to it. The fuzzy decision manager then looks up the profile associated with the web application. A profile is a collection of statistics associated with one specific web application. The fuzzy decision manager present in the profile is a set of keys and scores used to evaluate the features of a query and operate in one of two modes, learning or detection. In the learning phase we use C4.5 classification technique which models and builds a profile of the “normal” characteristics of a given feature of a query (e.g., the normal lengths of values for attributes), setting a dynamic detection threshold for the attributes. During the detection phase, models return an anomaly score for each observed example of attribute values using the fuzzy rules generated from the training phase.

This is just a probability value in the range of 0-1 indicating the degree of anomaly of the observed value in relation to the existing profile for that query which is computed using a fuzzy logic decision manager.

Access control manager: The architecture of this anomaly detection system necessitates the existence of an access control manager between the query classification component and web servers. This manager is utilized when a malicious web request that was let through by the query classifier to access the web server which can be checked for privilege level using anomaly score. In this system, access control can choose to update privilege levels of the web request to control malicious requests. This process involves characterizing the incoming anomaly using fuzzy rules and then generating updating messages and finally updating the access privilege levels to reflect the level of anomaly. In this three access levels namely privilege user lever, application programmer level and naïve user level are used. Queries with privilege user and application programmer level are sent to the smart server where as the queries with naïve user levels are sent to dump server.

Feedback propagation module: The feedback propagation module maintains profile for each user containing query details such as attributes details executed by them and the number of requests provided by them in a time interval, the location from which queries are sent, parameters passed in procedures executed by the application and the tokens used. These profiles are updated after each query is updated using anomaly scores and rules. The feedback is sent in the form of rules to the fuzzy anomaly detection module so that further queries by the same user are analyzed and monitored effectively.

Web servers: Two types of web servers are used in this architecture namely a smart server and a dumb server. Important queries with high privilege coming from privilege users and application programmers are executed by the smart server. Less privilege queries sent by attackers are treated as naïve user queries and are executed by dumb server.

Web database: This web database consists of three major components namely data dictionary encrypted (important) data and public data. The encrypted data is divided as two sub groups namely highly confidential data and confidential data. Highly confidential data stored in encrypted using elliptic key cryptography algorithm and the confidential data are encrypted using DES algorithm. The public data are stored without encryption.

Experimental Evaluations: Our approach uses several different evaluations about requests, parameters and their relationship to detect anomalous entries. An evaluation is a set of statistical and fuzzy rule based procedures used to evaluate a certain feature of a request. A feature can be related to parameters of a query string (e.g., the string length of a particular parameter values), or to some relationships between a request and others related to a specific web page or script r (e.g., number of requests in a time slot) and the spatio-temporal features We used five feature evaluations and different statistical tools:

- Length of parameter values
- Distribution of characters in parameter values
- Token finder

Learning: The learning phase works for establishing the normal behavior on the base of training set using a Neuro-fuzzy learning technique. For each feature, we estimate the corresponding probability distribution. The learning phase has the goal of setting a detection threshold. This task is performed in three steps: First, we set a weight (W_i) for each feature. The W_i values are established a priori and, if needed, they can be adjusted by the security manager based on constraints after a brief analysis process on historical data of web server as well as the spatial features of users. Then, for each web page/script (profile P), we compute the normal score (AS(r)) of each request r in the training set (Vigna *et al*., 2009) using (1):

$$\text{Anomaly score} = \sum\nolimits_{i \varepsilon \text{modles}} Wi * (1 - Pi) \qquad (1)$$

Finally, for each web page/script (profile), we set the initial anomaly score threshold, T_o (P) as the value of highest anomaly score.

Detection: In the detection phase, we compute the anomaly score of a given request and compare it with the score obtained in the training phase. If the anomaly score is higher, then the behavior is far from the normal behavior. The Anomaly Score (AS(req)) for the current request to the web page or script with profile is computed using the above formula. Then the anomaly

detection module takes a decision according to the following rules:
When AS(req) < T0(P), req is a normal request
When AS(req) > T0(P), an alert is sent

Learning and detection for length of the attribute: The length of http request attribute is used to detect anomalous request, when the input is abnormal. First, for the buffer overflow attacks in the target application, it is necessary to put the shell code and additional padding, according to the length of the target buffer. Second, for Cross Site Scripting attacks it attempts to include scripts in pages whose content is determined by user-supplied data which requires certain amount of data to be sent that significantly exceed the length of legitimate parameters. The goal of this model is to approximate the unknown distribution of the parameter lengths and detect instances that substantially deviate from the observed normal behavior.

Learning: We approximate the mean μ and the variance σ^2 of the actual length of attribute by calculating the mean μ and the variance σ^2 for the lengths $l_1, l_2, \ldots, l_n$ of the parameters processed during the learning phase. The length distribution is based on (Vigna *et al*., 2009) Eq. 2.

Detection: For the given attribute, the length is estimated with parameters μ and σ^2 as determined by the previous learning phase, it is the task of the detection phase to detect the anomalousness of a parameter with length l. To assess the anomaly of a string, we estimate the deviation of the length l from the mean value μ. The Chebyshev inequality puts, for an arbitrary distribution with variance σ^2 and mean μ, a first upper bound on the probability that the difference between the value of a random variable x and μ exceeds a certain threshold t:

$$p(l)\begin{cases} \dfrac{\sigma_2}{\sigma 2+(l-\mu)^2} & 1, l \geq \mu \\ & , l \geq \mu \end{cases} \tag{2}$$

The second and third upper bounds are decided by the spatio temporal features.

Learning and detection for character distribution: The attribute character distribution model examines the concept of a normal parameter by looking at its character frequency and its distribution even though the characters are stored in encrypted form the data are again in character form since we use the RSA encryption technique which encrypts characters with 64 bit keys to provide new characters and only the legimate users know the key for decryption. This approach is based on the observation that attributes have a regular structure and they almost contain only human readable characters. This becomes evident when the relative frequencies for all possible 256 characters in a string are sorted in descending order. The relative frequencies of character sorted in decreasing order are called its character distribution. The character distribution of an attribute that is normal which is non anomalous is called the attribute's Idealized Character Distribution (ICD).

Learning: The learning phase occurs as follows. For each HTTP request, compute the relative frequency of each character in the query section of the request and sort the characters in order of decreasing frequency and add the sum of the relative frequencies for each group to the cumulative total for each group. Once all requests have been processed the cumulative total for each group is divided by the number of requests, in order to obtain the average value for each group. These average values are referred to as the expected values which are used during the application of the Chi-square test in the detection phase. The sorted relative frequencies obtained above are placed into 6 groups based on the frequency of the characters seen in the query section in descending order. The first group contains the frequency of the most frequent character. The second and third groups contain the sum of the next three most frequent characters and the fourth group contains the sum of the next five. The fifth group contains the sum of the next four and the final group is the sum of the frequencies of remaining characters.

Detection: In the detection phase a modified version of the chi-square statistical test is used to identify anomalous requests. The Chi-square test computes the probability (or confidence level) that the character distribution values seen during testing are representative of the expected values computed during the training phase. The steps in applying the chi-square test to an HTTP request during the testing phase are as follows:

Calculate the observed values, i.e., the sorted, grouped character distribution of the query component of the request. Adjust the expected values computed during training by multiplying them by the length of this request. Calculate the chi-square (Vigna *et al*., 2009) value using Eq. 3, where O_i and E_i represent the observed and expected values, respectively:

$$\chi^2 = \sum_{i=0}^{6} \frac{(Oi - Ei)2}{Ei} \tag{3}$$

Based on the value of χ^2 and the degrees of freedom we compute the probability. The anomaly score is simply one minus this probability. If the anomaly score is above a particular threshold an alert is triggered. The threshold for all models was set to be the highest anomaly score seen during training plus 10% of that value.

Learning and detection for token finder: The purpose of the token finder model is to inspect, whether the values of an attribute in a query is from a limited set of possible choices. Web applications often require one out of a few alternatives for certain query attributes, such as flags. When anomalous user attempts to pass illegal values to the application, the attack can be detected. When no enumeration can be identified, it is assumed that the attribute values are random.

Learning: The classification of an argument as normal or random is based on the reflection of the number of different occurrences of parameter values is constrained by unknown threshold t. When the number of attributes increases proportional to the total number of attribute instances, then the use of random value is indicated. If such an increase is not observed, we assume an enumeration. Formally, to decide if attribute a is an enumeration, we compute the statistical correlation ρ between the values of the functions f and g for numbers 1....i of occurrences of *a*. The functions f and g are defined as follows in (Vigna *et al*., 2009) Eq. 4.

For F(x) = x, then:

$$g(x)\begin{cases} g(x-1)+1, \text{if a is new} \\ g(x-1)-1, \text{if a is not new} \, 0, \text{ if} x=0 \end{cases} \tag{4}$$

The correlation parameter ρ is obtained after the training data has been processed. It is calculated by using f and g with their variances Var(f), Var(g) and the covariance Covar (f, g) as shown below (Vigna *et al*., 2009):

$$\rho = \frac{\text{Co}\,\text{var}(f,g)}{\sqrt{\text{Var}(f)-\text{Var}(g)}} \tag{5}$$

If correlation < 0,

Then f and g are correlated negatively and enumeration is assumed. This is incited by the fact that, increasing value of f correlates with decreasing values of g(x).

In the reverse process, where ρ > 0, the attributes have shown significant variation to support the possibility that it is not from the set of predefined tokens. When an enumeration is assumed, the complete set of identifiers is used in the detection phase.

Detection: Once it has been determined that the values of a query attribute is an enumeration, any new value is expected to appear in the set of known values. When this happens, 1 is returned, 0 otherwise. If it has been determined that the parameter values are random or anomalous, the model returns 1.

MATERIALS AND METHODS

The System was implemented in java and we used C4.5 classifier algorithm and intelligent fuzzy rules for the detection of various types of attacks. Our system SQL anomaly detection system works fine and proved to be efficient.

RESULTS

This system was tested with queries which results in considerable improvement over the existing system. The system's capability is the reduction of false positives and has been proved using new approaches in addition to providing privileged access to web sites based on the anomaly score associated with web requests.

DISCUSSION

Table 1 compares the number of anomalies detected in our system with existing system. The intelligent SQL anomaly detector detects all types of anomalies in our system with the use of C4.5 classification algorithm which classifies based on user queries sent to the system. Our system proves that it detects all anomalies and shows better results when compared with the existing system.

Table 1: Number of anomalies detected from the queries sent

No. of queries sent	No. of anomalies detected in existing system	No. of anomalies detected in our system
200	155	187
300	237	280
400	305	373
500	371	458
600	496	586

Table 2: Number of malicious detections and preventions from malicious attempts

No. of malicious attempts	No. of malicious detections in existing system	No. of malicious detections in our system
50	39	46
100	71	88
150	113	134
200	146	179
250	175	235

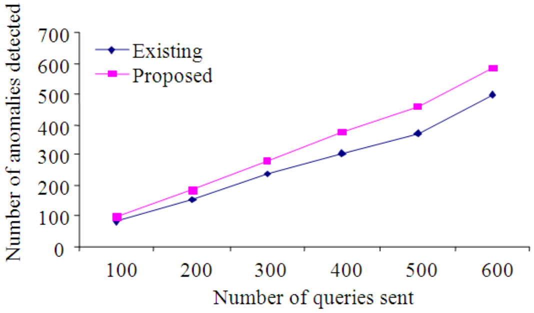

Fig. 2: Number of anomalies detected in our system is compared with existing system

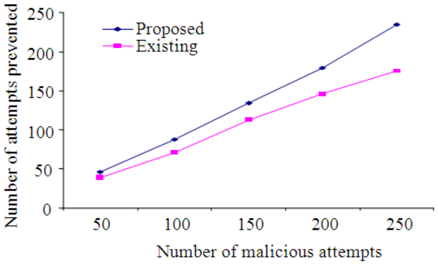

Fig. 3: Number of malicious attempts Prevented in our system is compared with existing system

Figure 2 shows the number of anomalies detected based on the number of queries sent in our system which is compared with the existing system. This graph proves that our system detects more number of anomalies than the present system.

Table 2 compares the number of malicious detections and preventions with respect to the number of malicious attempts. The number of detections and preventions of malicious attempts is more when compared to the present system. Since, it uses intelligent SQL anomaly detection system with combined analysis of C4.5 classifier algorithm and intelligent fuzzy rules. Our system proves that it detects and prevents all malicious attempts and shows better results when compared with the existing system.

Figure 3 shows the number of malicious detections and preventions in which it compared with the present system. This graph proves that our system detects and prevents more number of malicious attempts than the existing system.

CONCLUSION

In this study, a refreshing technique to improve the detection rate of web-based intrusion detection systems by serially framing a web request anomaly detector using fuzzy rules has been proposed and implemented. This model allows one to route anomalous requests to servers that have limited access to sensitive information. This system reduces the damage in the situation of an attack and simultaneously, maintains some level of service to anomalous queries that do not require access to sensitive information. The main advantage of this system is the prevention of attackers and allowing legitimate users to retrieve information based on keys. Further studies in this direction can be inclusion of misuse detection feature into this system.

REFERENCES

Almgren, M., H. Debar and M. Dacier, 2000. A lightweight tool for detecting web server attacks. Proceeding of the ISOC Symposium on Network and Distributed Systems Security, (SNDSS'2000), San Diego, pp: 157-170.

Syurahbil, N. Ahmad, M.F. Zolkipli and A.N. Abdalla, 2009. Intrusion preventing system using intrusion detection system decision tree data mining. Am. J. Eng. Applied Sci., 2: 721-725. DOI: 10.3844/ajeassp.2009.721.725

Mehdi, M., S. Zair, A. Anou and M. Bensebti, 2007. A Bayesian networks in intrusion detection systems. J. Comput. Sci., 3: 259-265. DOI: 10.3844/jcssp.2007.259.265

Sundararajan, T.V.P. and A. Shanmugam, 2010. A novel intrusion detection system for wireless body area network in health care monitoring. J. Comput. Sci., 6: 1355-1361. DOI: 10.3844/jcssp.2010.1355.1366

Toth, T. and C. Kruegel, 2002. Accurate buffer overflow detection via abstract pay load execution. Lect. Notes Comput. Sci., 2516: 274-291. DOI: 10.1007/3-540-36084-0_15

Vigna, G., F. Valeur, D. Balzarotti, W. Robertson and C. Kruegel *et al*., 2009. Reducing errors in the anomaly-based detection of web-based attacks through the combined analysis of web requests and SQL queries. J. Compt. Security, 17: 305-329, DOI: 10.3233/JCS-2009-0321