Analyzing HTTP requests for web intrusion detection

Sara Althubiti
North Carolina A & T State University, saalthub@aggies.ncat.edu

Xiaohong Yuan
North Carolina A & T State University, xhyuan@ncat.edu

Albert Esterline
North Carolina A & T State University, esterlin@ncat.edu

Abstract
Many web application security problems related to intrusion have resulted from the rapid development of web applications. To reduce the risk of web application problems, web application developers need to take measures to write secure applications to prevent known attacks. When such measures fail, it is important to detect such attacks and find the source of the attacks to reduce the estimated risks. Intrusion detection is one of the powerful techniques designed to identify and prevent harm to the system. Most defensive techniques in Web Intrusion Systems are not able to deal with the complexity of cyber-attacks in web applications. However, machine learning approaches could help to detect known and unknown web application attacks. In this paper, we present machine learning techniques to classify the HTTP requests in the well-known dataset CSIC 2010 HTTP (Giménez et al., 2012) as normal or abnormal traffic, and we compare our experimental results with the results reported by Pham et al. in 2016 and Nguyen et al. in 2011. These experiments produce results for overlapping sets of machine-learning techniques and different sets of features, allowing us to compare how good the various feature sets are for the various machine-learning techniques, at least on this dataset.

Keywords: intrusion detection system; anomaly detection; web application attacks; machine learning.

Disciplines
Other Computer Sciences

1. Introduction

Web servers and Web applications are widely used in various organizations, and they have been targeted by numerous attacks that may cause huge damage to the system. To reduce the risk of web application attacks, web application developers need to write secure applications to prevent known attacks. When the secure application fails, it is important to detect such attacks. Attack detection is important for incident response, limiting the damage of attacks, prosecuting the attacker, deterring attacks, and prevention of future attacks.

Intrusion detection is one of the powerful technique designed to identify and prevent harmful activities on a system (Khan et al., 2016). Intrusion detection has two main classes: misuse detection and anomaly detection. Misuse detection attempts to identify instances of web application attacks by comparing current activity against the expected actions of an attacker, usually by using pattern-matching algorithms. In contrast, an anomaly detection approach studies the behavior of the user, whether a client or a server, and detects whether the behavior is normal or anomalous, often using machine learning techniques. Existing anomaly web intrusion detection approaches include several techniques based on statistical models for characterizing query parameters (Kruegel&Vigna,2003), feature-based data clustering (Das et al., 2009), anomaly detection by using rule sets (Auxilia et al., 2010), learning the profiles of normal database access performed by web-based applications (Valeur et al., 2005), and others. These approaches have been used to detect such attacks as SQL injection, cross-site scripting, distributed denial of service, HTTP attacks, and so on.

Machine learning techniques allow one to implement an anomaly detection system that can learn from training (labeled) data and provide the decision for test (unlabeled) data (Singh et al.,2013). To use machine learning classification algorithm to classify HTTP requests as normal or anomalous, first extract features from the row data and label the data based on these features, each instance has multiple features and one label(class). By learning how the features relate to the label, a mathematical model will be produced that maps the relationship between features and labels. That model, is known as the classifier and utilized to predict the class of each record in the test data.

In this paper, we classify HTTP traffic as normal and abnormal by applying a set of machine learning techniques, and we compare the experimental results with those obtained by (Pham et al., 2016) and (Nguyenet et al., 2011). In order to gain good machine learning performance, we took the nine features used in (Nguyenet et al., 2011), ranked them using the attribute evaluator methods that are built into Weka (Hall et al., 2003), and then kept only those five that improved the learning results.

The rest of this paper is organized as follows. After the Introduction section, Section 2 describes related work, Section 3 presents experiments and results, a discussion of the findings is presented in Section 4, and conclusions and future work are presented in the last section.

2. Related work

Enhancing intrusion detection with machine learning has been done before. (Pham et al., 2016) surveyed different machine learning algorithms such as random forest, logistic regression, decision tree, AdaBoost, and SGD that are used to build Web intrusion detection systems. Moreover, the authors built an experimental framework for comparing the performance of some machine learning techniques running on the CSIC 2010 HTTP dataset (Giménez et al., 2012), which contains generated traffic targeted to an e-commerce Web application. Their results suggested that logistic regression is the best learning technique for this problem among the techniques investigated. Logistic regression provided a decent performance with the highest recall and highest precision.

In addition, (Nguyenet et al., 2011) proposed a framework to utilize the generic feature selection (GeFS) measure for Web intrusion detection. For intrusion detection, they applied the GeFS method together with two measures that are coupled with search strategies: the correlation feature-selection (CFS) measure and the minimal-redundancy-maximal-relevance (mRMR) measure. GeFS is generally used to select features from high-dimensional datasets, such as network traffic or web logs. This technique allows one to evaluate feature subsets not only by their relevance, but also by the relationships between features. CFS identifies the relevance of features and their relationships in terms of linear correlation, and mRMR selects features from datasets that have many non-linearly correlated features They analyzed statistical properties of the newly generated CSIC 2010 HTTP dataset and the ECML/PKDD 2007 dataset (Gallagher et al., 2009). The detection accuracies obtained after feature selection were calculated as the average of four different classifiers. Their result showed that CSF achieved good performance on the CSIC 2010 dataset while mRMR performed well on the ECML/PKDD 2007 dataset, which is a collection of real-world web traffic. The data was portioned into a training set and a test set. The training data was made available to challenge participants. The test set was released only once the Discovery Challenge was complete.

(Yu et al., 2016) performed hybrid intrusion detection based on anomaly detection and misuse detection as revealed in Web logs. Their model enjoys the advantages of both the anomaly detection model based on a clustering algorithm and the misuse detection model, which is rule based. Malicious log records that cannot be detected by the misuse detection model are loaded into the anomaly detection

model for a second attempt at detection.

Moreover,(Zolotukhin et al., 2014) considered how HTTP logs could be analyzed for network intrusion detection. When a training set of HTTP requests that does not contain any attacks is analyzed and all relevant information has been extracted from the logs, clustering and anomaly detection techniques are applied to define a model of normal user behavior. The model was used to identify network attacks as deviations from the normal in an online mode**.**

(Fan and Guo, 2012) proposed an adaptive model that detects Web-based attacks by recognizing normal traffic and utilizing several hidden Markov models. Through interpreting the structural features of an HTTP request message, they extract the destination URL, which is a string in standardized format used to identify the location of a resource on the Internet. The log file data was divided into a few smaller sets according to request type. The differentiation of subsets was determined by several properties such as date, host, and referrer headers, IP address, and port number. Analyzing how one may differentiate Web requests to decide whether a request is normal, they were able to build a detector based on a hidden Markov model. The experimental outcomes demonstrated that the adaptive model can successfully recognize Web-based attacks and reduce false alerts.

Finally,(Kruegel et al., 2003) presented an intrusion detection system that uses various distinctive anomaly detection strategies to detect attacks against Web servers and Web-based applications. The system associates the server-side programs referenced by client queries with the parameters contained in these queries. The specific characteristics of the application of the parameters enable the system to perform attentive analysis and deliver a reduced number of false positives.

3. Experiment

This section presents the experimental procedures for and results of applying various machine learning techniques to the CSIC 2010 HTTP dataset. For applying these techniques, we used the machine-learning tools available in Weka. Weka (Waikato Environment for Knowledge Analysis) is a machine learning tool (Witten et al., 1999). We used attribute evaluator methods in Weka to rank the nine features used in (Nguyenet et al., 2011) and used the best five in our applications(see Table 2), which gave better results compared to (Pham et al., 2016) and (Nguyenet et al., 2011).

3.1 Datasets

The experiment was conducted on the CSIC 2010 HTTP dataset (Giménez et al., 2012), which contains generated traffic targeted to an e-commerce Web application. The resulting dataset contains 36,000 normal requests and more than

25,000 abnormal requests. In this data, the requests are labeled as normal or abnormal and include several attacks, such as SQL injection, buffer overflow, information gathering, file disclosure, CRLF injection, XSS, and so on.

3.2 Feature selection

Feature selection is the process of selecting the most relevant attributes to classify the data. A simple example of this process is the following: If you are trying to determine whether a person is happy, a potential feature is whether that person is smiling or not. Reading through (Nguyenet et al., 2011), we found nine features listed that we considered important for the detection process (see Table 1). We used feature selection methods in Weka to rank these features and used the best five in our application to improve the accuracy and decrease the training time (see Table 2). Feature selection process in Weka contains two methods, attribute evaluator method and search method. The attribute evaluator is a technique that shows how each attribute in the dataset is assessed in the context of the output, while, the search method, represents how the attributes could be navigated or explored in the dataset (Hal et al., 2009). In our model, "WrapperSubsetEva" has been used as an attribute evaluator method to assess the attributes using J48 classifier and 10-fold cross validation. "BestFirst" was used as a search method to navigate the attribute subsets. The best five features were ranked based on their importance and impact on the accuracy (see Table 2). Some features refer to the length of the arguments, the length of the request, the length of the path or the headers as length is a significant factor for detecting buffer-overflow attacks. Also, we found that there are special characters in numerous injection attacks. We studied their occurrence in the path and in the arguments' values.

Table 1. ***Names of 9 features that are considered relevant for the detection of Web attacks in the CSIC-2010 HTTP dataset.***

Feature Name
Length of the request
Length of the arguments
Number of arguments
Number of digits in the arguments
Length of the path
Number of letters in the arguments
Number of letter chars in the path
Number of 'special' chars in the path
Maximum byte value in the request

Table 2. ***Names of 5 features that are scored important for the detection of Web attacks in the CSIC-2010 HTTP dataset.***

Feature Name
Length of the request
Length of the arguments
Number of arguments
Length of the path
Number of 'special' chars in the path

3.3 Experimental settings

We compare different machine learning techniques, including random forest, logistic regression, AdaBoost, J48 (a decision tree technique which includes CART and C4.5), SGD (stochastic gradient descent),and Naïve Bayes in order to identify the difference in performance in terms of the accuracy rate between our study and (Pham et al., 2016) and (Nguyenet et al., 2011). The dataset was divided into 60% as a training set and 40% as a test set.

3.4 Experimental Results

The performance of each method is measured by its precision, recall, F-Measure, TP rate and FP rate on the test set. These measures for our set of features for each method are shown in the (Table 3.a). In the following, 'TP' and 'TN' refer to the number of true positives and negatives, respectively, and 'FP' and 'FN" refer to the number of false positives and negatives, respectively. P = TP + FN, the number of (possibly misclassified) positive observations, and N = TN + FP, the number of (possibly misclassified) negative observations.

Precision is the proportion of positive predictions that are correct:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Recall is the proportion of all positive observations that are classified as such:

$$Recall = TPR = \frac{TP}{TP + FN} = \frac{TP}{P} \quad (2)$$

The F-Measure is a measure of the test's accuracy, it is the harmonic mean of precision and recall:

$$F\text{-}Measure = 2* \frac{Precision*Recall}{Precision+Recall} \quad (3)$$

The FP rate is defined as the proportion of all negative observations that are classified as such

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N} \quad (4)$$

And Detection Rate (accuracy) is proportion of all corrected prediction

$$Detection\ Rate = \frac{TP + TN}{TP + FN + TP + TN} \quad (5)$$

Table3. Various metrics for various machine-learning techniques run on the CSIC 2010 HTTP dataset across several sets of features

*Table 3**.a Our experimental results***

Methods	RF	LR	J48	ABc	SGDc	NB
Detection Rate	99.94	99.94	99.94	99.94	99.88	88.83
Precision	99.90	99.90	99.90	99.90	99.90	89.00
Recall	99.90	99.90	99.60	99.90	99.90	88.80
F-Measure	99.90	99.90	99.80	99.90	99.90	88.90
TP Rate	99.90	99.90	99.60	99.90	99.90	88.80
FP Rate	00.10	00.10	00.10	00.10	00.20	11.00

RF=Random Forest, LR=Logistic Regression, J48=Decision Tree , ABc= AdaBoost Classifier, SGDc= Stochastic Gradient Descent Classifier, and NB=Naïve Bayes .

*Table 3.b **(Pham et al., 2016) results***

	Methods	RF	LR	DT	ABc	SGDc
anomalous	**Precision**	79.70	99.39	88.10	67.24	72.45
	Recall	87.11	93.05	88.28	89.19	92.04
	F1 score	83.24	96.11	88.19	76.68	81.08
normal	**Precision**	83.37	92.54	86.48	80.06	86.69
	Recall	74.46	99.34	86.26	49.98	59.70
	F1 score	78.67	95.82	86.37	61.54	70.71

RF=Random Forest, LR=Logistic Regression, DT=, ABc=AdaBoost Classifier, and SGDc=Stochastic Gradient Descent Classifier.

Table 3**.c (Nguyenet et al., 2011) results**

	Methods	RF	C4.5	CART	RT
Full-Set	**Detection Rate**	93.71	94.49	94.12	92.30
	FP Rate	7.2	5.9	6.2	8.3
CFS	**Detection Rate**	93.68	94.06	93.71	92.70
	FP Rate	7.2	6.8	6.8	7.8
mRMR	**Detection Rate**	71.70	79.80	79.85	71.36
	FP Rate	30.5	25.7	25.3	30.6

RF=Random Forest, C4.5=Decision Tree, CART=Classification and Regression Trees, and RT=Random Tree. CFS=Correlation Feature-Selection, and mRMR=Minimal-Redundancy-Maximal-Relevance.

Table 3.a, Table 3.b and Table 3.c all are shows the experimental results of applying various machine learning methods on CSIC 2010 HTTP dataset but with different features sets and different measures.

Precision and high recall, where high precision correlates to a low false positive rate, and high recall correlates to a low false negative rate. All proposed techniques are good and have decent performance in this kind of problem because

here we have a binary nominal classification and the attributes or features are numeric. A high recall and low precision technique proceed numerous outcomes, but most of its predicted labels are inappropriate once compared to the training labels. On the other hand, high precision and low recall technique yield to limited outcomes, but accurate predicted labels once compared to training labels. Nonetheless, an ultimate system, high precision and high recall, proceed many results that are labeled properly (Makhoul et al.,1999). All methods achieved high detection rate, high precision and high recall and low FPR.

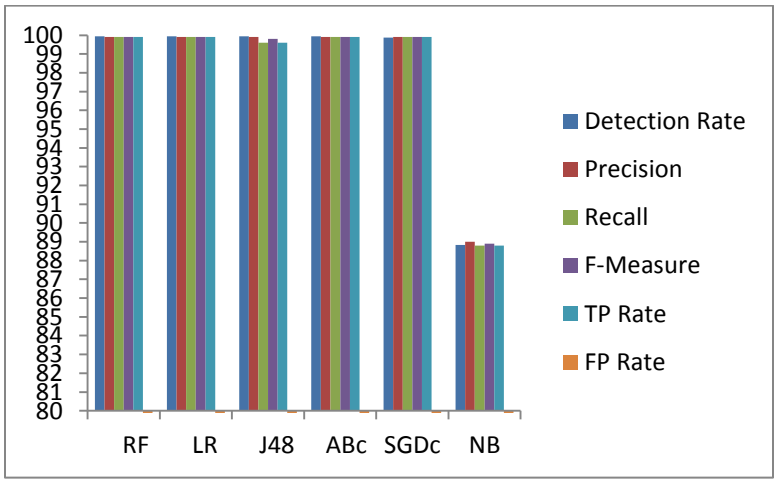

Figure 1 .Graph of Detection rate, Precision, Recall, F1-Measure, TPR and FPR of various learning techniques with our set of features.

Figure 1 is a graph of detection rates, precision, recall, F-measures, TPR and FPR of the machine learning algorithms on CSIC 2010 HTTP dataset with our set of features.

4 Discussion of findings

The purpose of this study was to show how different sets of features could be effective with different machine learning techniques to classify HTTP requests as normal and abnormal traffic by applying them on the CSIC 2010 HTTP dataset. This study showed that all the techniques have high precision and recall, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate, except Naïve Bayes. The findings of this study are consistent with those of (Nguyen et al., 2011), where the extracted features (see Table 1) in both studies are similar. Even given the existing similarity, our study achieved somewhat better accuracy rates in all applied machine learning techniques because, in our study, we have used some of Weka's attribute evaluator methods to rank the features and we found that a subset (see Table 2)of the features used by (Nguyen at al.,2011) gave results superior to those obtained not only with their

full set of features but also with their two optimized subsets. In addition, (Pham et al., 2016) surveyed the results of various machine learning algorithms applied to the CSIC 2010 HTTP dataset but with a set of extracted features different from ours; our accuracy again was consistently higher. In summary, because we used attribute evaluator methods in Weka to rank the nine features used in (Nguyenet et al., 2011) and used the best five in our applications, we got better results compared to (Pham et al., 2016) and (Nguyenet et al., 2011).

5 Conclusions and future work

In this paper, different machine learning techniques were applied to the CSIC 2010 HTTP dataset for intrusion detection purposes. The dataset included attacks such as SQL injection, buffer overflow, information gathering, files disclosure and so on. Experiments showed that all techniques have high precision, recall, and F1-measures and low FPR, except Naïve Bayes which shows less precision, recall, and F1-measures and high FPR comparing to the rest of the techniques . (Nguyen et al., 2011) extracted nine features considered important for the detection process, and we used the best five as selected by Weka; this gave better results, high accuracy and cuts in the training time.

There is an abundance of potential research that may arise from this paper. First, one could evaluate the proposed methods on various other datasets. Secondly, one could apply semi-supervised machine learning techniques on this dataset and see how the performance for intrusion detection changes. Note that semi-supervised techniques can often give results comparable to supervised-learning techniques but require many few labeled training records and thus much less expensive labeling.

References

Giménez, C.T., Villegas, A.P., and Marañón, G.A., "HTTP Dataset CSIC 2010," CSIC (Spanish Research National Council), 2012, http://www.isi.csic.es/dataset/

Pham, T.S., Hoang, T.H. and Vu., V.C. "Machine learning techniques for web intrusion detection—A comparison." *Eighth International Conference on Knowledge and Systems Engineering (KSE),*. IEEE, 2016.

Nguyen, H.T., et al. "Application of the generic feature selection measure in detection of web attacks." *Computational Intelligence in Security for Information Systems*. Berlin: Springer, 2011, 25-32.

Yu, J., Tao, D., and Lin, Z. "A hybrid web log based intrusion detection model." *4th International Conference on Cloud Computing and Intelligence Systems (CCIS),* IEEE, 2016.

Zolotukhin, M., et al. "Analysis of http requests for anomaly detection of web attacks." *12th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC),* IEEE, 2014.

Fan, W.K.G.. "An adaptive anomaly detection of WEB-based attacks." *7th International Conference on Computer Science & Education (ICCSE), 2012*. IEEE, 2012.

Kruegel, C., and Vigna, G. "Anomaly detection of web-based attacks." *10th ACM conference on Computer and communications security*. ACM, 2003.

Nowson, S. "Scary films good, scary flights bad: Topic driven feature selection for classification of sentiment." 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion (TSA '09), ACM, 2009, pages 17–24

Khan, Javed Akhtar, and Nitesh Jain. "A Survey on Intrusion Detection Systems and Classification Techniques." *IJSRSET* 2.5 (2016): 202-208.

Das, D., Sharma, U.,and Bhattacharyya, D.K. "A Web Intrusion Detection Mechanism based on Feature based Data Clustering." IEEE International Advanced Computing Conference, IEEE, 2009, pp. 1124 – 1129.

Auxilia, M. and Tamilselvan, D. "Anomaly detection using negative security model in web application." International Conference on Computer Information Systems and Industrial Management Applications (CISIM), IEEE, 2010, pp. 481-486.

Valeur, F, Mutz, D., and Vigna, G. A. "Learning-based approach to the detection of SQL attacks", *Conference on Detection of Intrusions and Malware and Vulnerability Assessment* (DIMVA), Austria, 2005.

Singh, J. and Nene, M.J. "A survey on machine learning techniques for intrusion detection systems." *International Journal of Advanced Research in Computer and Communication Engineering* 2.11 (2013): 4349-4355.

Gallagher, B., and Eliassi-Rad, T. "Classification of http attacks: a study on the ECML/PKDD 2007 discovery challenge. Tech. Report No. LLNL-TR-414570. Lawrence Livermore National Laboratory, Livermore, CA, 2009.

Hall, Mark A., and Geoffrey Holmes. "Benchmarking attribute selection techniques for discrete class data mining." *IEEE Transactions on Knowledge and Data engineering* 15.6 (2003): 1437-1447.

Witten, Ian H., et al. "Weka: Practical machine learning tools and techniques with Java implementations." (1999).

Makhoul, John, et al. "Performance measures for information extraction." *Proceedings of DARPA broadcast news workshop*. 1999.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009).
The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10-18.