Learning System for SQL Injection Detection Using Syntax and Semantic Kernel in Support Vector Machine

Yi Wang[1] , Zhoujun Li[1]

[1] State Key Laboratory of Software Development Environment, Beihang University
100191 Beijing, China
{wangyi160@hotmail.com, lizj@buaa.edu.cn}

Abstract. Modern web application systems are generally consisted of database systems in order to process and store business information. These systems are highly interesting to hackers as they contain sensitive information and the diversity and amount of attacks severely undermine the effectiveness of classical signature-based detection. In this work we propose a novel approach for learning SQL statements and apply machine learning techniques, such as one class classification, in order to detect malicious behavior between the database and application. The approach incorporates the tree structure of SQL queries as well as input parameter and query value similarity as characteristic to distinguish malicious from benign queries. We develop the learning system integrated in PHP and demonstrate the usefulness of our approach on real-world application.

Keywords: sql injection, web security, machine learning, support vector machine, kernel tricks

1 Introduction

The majority of today's web-based applications does employ the multi-layer infrastructure and rely heavily on database storage for information processing. A lot of attacks against web-applications are aimed at injecting commands into database systems in order to gain unprivileged and access to sensitive records stored in these systems. The approach of protecting web application is by introducing detection models on the network layer firewall systems.

Besides pattern based approaches, there exists a variety of research on employing anomaly based methods for detecting web-based intrusions or program analysis on source code of target web application[1,2,3,4,5,6,7,8]. The main contribution of our work is the use of both syntax and semantic based analysis, i.e. tree-vector-kernel based learning, which became popular within the field of natural language processing (NLP). Our approach incorporates the parse tree structure of SQL queries as well as input parameter and query value similarity characteristic to distinguish malicious from benign queries. By applying this kernel trick into the SVM(support vector machine) classifier, we can determine abnormal query accurately and efficiently.

2 Kernel function for SQL query

The kernels between corresponding pairs of trees and/or vectors in the input sequence are summed together.

$$K_s(o_1,o_2)=\tau\times\sum_{i=1,\ldots,min(n,n')}k_t(T_i,T_i')+\sum_{i=1,\ldots,min(n_v,n_u)}k_b(\vec{v}_i,\vec{u}_i) \quad (1)$$

2.1 Tree kernel function

The main idea of tree kernels is to compute the number of the common sub-structures between two trees *T*1 and *T*2 without explicitly considering the whole fragment space. For this purpose, we need to define the tree kernel function in order to compute the similarity of two trees.

$$K_t(T_1,T_2)=\sum_{n_1\in N_{T_1}}\sum_{n_2\in N_{T_2}}\Delta_k(n_1,n_2) \quad (3)$$

where N_{T_1} and N_{T_2} are the sets of the T_1's and T_2's nodes, respectively. By adopting the concept of tree kernel from (ECAL 2006), we can define

$$\Delta_k(n_1,n_2)=\begin{cases}0 & if\ prod(n_1)\neq prod(n_1)\\ \lambda & if\ height(n_1)=height(n_2)=1\\ \lambda\prod_{j=1}^{|n_1|}(\sigma+\Delta(c_{n_1}^j,c_{n_2}^j)) & otherwise\end{cases} \quad (4)$$

where λ is the decay factor and $\sigma\in[0,1]$ is the counting factor, $|n|$ is the number of the children of node, for the last condition, $|n_1|=|n_2|$.

2.2 Vector kernel function

The best-known character-based string similarity metric is Levenshtein distance(LD). In order to make measurement for similar strings bigger than different strings, we define:

$$\Delta_b(u,v)=\frac{1}{LD(u,v)} \quad (5)$$

By making input-query value pair $P=\{\Delta_b(u_1,v_1),\Delta_b(u_2,v_1),\ldots,\Delta_b(u_m,v_n)\}$, we can define vector kernel to calculate the similarity:

$$K_b(P,P')=\sum_{i=1}^{|P|}Gaussian(P_i,P_i') \quad (6)$$

Where |P|=|P'| Gaussian(…) is the Gaussian radial basis function:

$$Gaussian(x_i,x_j)=exp(-\gamma\|x_i-x_j\|^2),\gamma>0 \quad (7)$$

3 System design and evaluation

We present our prototype system SQLLEARN as a mysqlnd extension integrated in PHP interpreter. It functions as a SQL proxy with the ability of query learning and anomaly detection between PHP application and Mysql database. Several vulnerable PHP content management system applications are tested within this framework, the results show that our system can provide accurate and complete protection against SQL injection attacks.

Moreover, we compare the tree-vector kernel with tree and vector kernel alone to show that the combination kernel surpass any singleton kernel and obtain the best result. This reflect the fact that both syntax and application context play important role in the detection of malicious SQL injection.

4 Conclusion

We presented an approach using tree-vector-kernels in SVM for SQL statements to prevent SQL injection in web applications. The results confirm the benefit of incorporation of syntax information of query and semantic context from application in analyzing SQL queries. Compared to previous approaches, the combination gains more accuracy than using syntax or context analysis alone as it brings more information into classification.

References

1. D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. Journal of Machine Learning Research. (2003).
2. Alessandro Moschitti, Making tree kernels practical for natural language learning. In Proceedings of the Eleventh International Conference on European Association for Computational Linguistics, Trento, Italy, (2006).
3. Lee, S.-Y., Low, W.L., Wong, P.Y.: Learning fingerprints for a database intrusion detection system. In: Gollmann, D., Karjoth, G., Waidner, M. (eds.) ESORICS 2002. LNCS, vol. 2502, pp. 264–280. Springer, Heidelberg (2002)
4. Buehrer, G., Weide, B.W., Sivilotti, P.A.G.: Using parse tree validation to prevent sql injection attacks. In: Proc. of SEM, pp. 106–113. ACM, New York (2005)
5. Gerstenberger, R.: Anomaliebasierte Angriffserkennung im FTP-Protokoll. Master's thesis, University of Potsdam, Germany (2008)
6. D¨ussel, P., Gehl, C., Laskov, P., Rieck, K.: Incorporation of application layer protocol syntax into anomaly detection. In: Sekar, R., Pujari, A.K. (eds.) ICISS 2008. LNCS, vol. 5352, pp. 188–202. Springer, Heidelberg (2008)
7. C. Bockermann, M. Apel, and M. Meier, "Learning sql. for database intrusion detection using context-sensitive modelling," in Detection of Intrusions and Malware, and Vulnerability Assessment, Volume 5587/2009. Springer Berlin / Heidelberg, pp. 196-205. (2009)
8. Ryan Dewhurst. Damn Vulnerable Web Application (DVWA). http://www.dvwa.co.uk/, (2012).