**Machine Learning Techniques for Cyber**

**Attacks Detection**

**Rafał Kozik1 and Michał Choraś1*,*2**

1

Institute of Telecommunications, UT&LS Bydgoszcz, Poland chorasm@utp.edu.pl

2

ITTI Ltd., Poznań, Poland [mchoras@itti.com.pl](mailto:mchoras@itti.com.pl)

Summary: The increased usage of cloud services, growing number of users, changes in network infrastructure that connect devices running mobile operating systems, and constantly evolving network technology cause novel challenges for cyber security that have never been foreseen before. As a result, to counter arising threats, network security mechanisms, sensors and protection schemes have also to evolve in order to address the needs and problems of nowadays users.

# Introduction

Recently there is an increasing number of security incidents reported all over the world. This situation is strongly related to the fact that recently there is also an increasing number of mobile devices users that form the population of connect-from-anywhere terminals that regularly test the traditional boundaries of network security.

In our previous work [2], we have introduced an innovative evolutionary algorithm for modeling genuine SQL queries generated by web-application. In this paper we have extended our algorithm with Bayes inference in order to incorporate advantages of signature-based and anomaly-based methods. The proposed approach allows for extracting patterns (in form of a PCRE regular expression) of a genuine SQL queries that can be easily incorporated in any rule processing engine (e.g. Snort). Moreover, the results showed that combining that kind of attack detector with character distribution allows for additional effectiveness improvements.

The paper is structured as follows. In Section 2 we present an overview of existing machine learning techniques for cyber attack detection. In Section 3 we present our own solution for SQL Injection attempts detection based on Bayes inference. The experimental setup and results are provided in consecutive sections. Conclusions are given thereafter.

R.S. Choraś (ed.), Image Processing and Communications Challenges 5, Advances in Intelligent Systems and Computing 233, DOI: 10.1007/978-3-319-01622-1\_44, © Springer International Publishing Switzerland 2014

# Overview of Methods for Cyber Attack Detection

## Signature-Based Methods

The Signature-based category of cyber attacks detection methods typically include Intrusion Prevention and Detection Systems (IDS and IPS) which use predefined set of patters (or rules) in order to identify an attack. The patterns (or rules) are typically matched against a content of a packet (e.g. TCP/UDP packet header or payload). Commonly IPS and IDS are designed to increase the security level of a computer network trough detection (in case of IDS) and detection and blocking (in case of IPS) of network attacks.

One of the most popular IDS/IPS software, widely deployed worldwide, is Snort [7]. Since it is an open source project, its users are allowed to freely modify it as well as feed the Snort engine with rules obtained from different sources (e.g. not only form Snort homepage).

Commonly the signatures (in form of reactive rules) of an attack for a software like Snort are provided by experts form a cyber community. Typically, for deterministic attacks it is fairly easy to develop patterns that will clearly identify particular attack. It often happens when given malicious software (e.g. worm) uses the same protocol and algorithm to communicate trough network with command and control center or other instance of such software. However, the problem of developing new signatures becomes more complicated when it comes to polymorphic worms or viruses. Such software commonly modifies and obfuscates its code (without changing the internal algorithms) in order to be less predictive and hard to detect.

Therefore, recently a machine learning based algorithms have been adapted for developing signatures that will be efficiently identify both code and behaviour of malicious code. The Network-based Signature Generation (NSG) [3], Length-based Signature Generation (LSEG) [4], and F-Sign [5] are examples of algorithms designed for automated and fast extraction of signatures of polymorphic worms. The LESG algorithm targets those worms that use buffer overflow attack to infect victims, while the F-Sign extracts the signature on a basis of code of a worm (such signature can be use to detect and stop worm from spreading). In literature there are also algorithms such as SA (Semantic Aware [6]) that are designed to generate signatures of malicious software on a basis of network traffic they generate. Such solutions like [6] can even properly identify malicious behaviour when the traffic is noise-like.

## Anomaly-Based Methods

The anomaly-based methods for a cyber attacks detection typically build a model that is intended to describe normal and abnormal behaviour of network traffic. Commonly such methods use two types of algorithms borrowed from machine learning theory, namely unsupervised and supervised approach.

For unsupervised learning commonly [8, 9, 10, 11, 12, 13, 14] clustering approaches are used that usually adapt algorithms like k-means, fuzzy c-means, QT, and SVM. The clustered network traffic established using mentioned approaches commonly requires decision whenever given cluster should be indicated as a malicious or not. Pure unsupervised algorithms use a majority rule telling that only the biggest clusters are considered normal. That means that network events that happen frequently have no symptoms of an attack. In practice, it is a human role to tell which cluster should be considered as an abnormal one.

The supervised machine learning techniques require at least one phase of learning in order to establish the traffic model. The learning is typically off-line one and is conducted on specially prepared (cleaned) traffic traces. One of the exemplar approaches to supervised machine learning for cyber attack detection uses auto regression stochastic process (AR) [15, 16, 17]. In literature there are also methods using Kalman filters [21]. Recently, more gaining in popularity are solutions adapting SVM [20], neural networks [19], and ID3-established decision trees [18].

# Proposed Approach to Injection Attack Detection

The proposed approach engages a Bayesian inference theory for cyber attacks detection. For that purpose a directed acyclic network (graph) is built, which is a graphic representation of the joint probability distribution function over a set of variables. In such graph each node represents random variable while the edge indicates a dependant relationship.

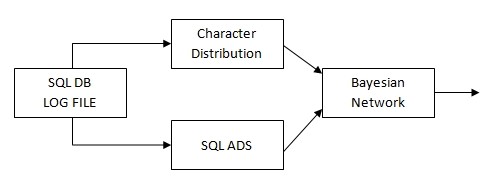


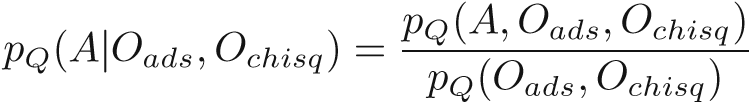
Fig. 1. Diagram of proposed algorithm

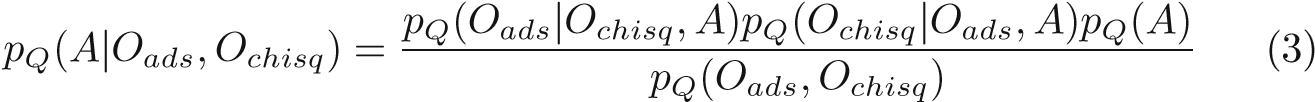
As it is shown in Fig.1, two types of observations are used in order to evaluate a probability of an attack. One of the observations is obtained from SQL-ADS, while the other comes from character distribution model. Both of the methods of injection attack detection are detailed in following sections. The fact that particular query (*Q*) is an injection attack (*A*) given the observation (*Oads*, *Ochisq*), is evaluated using formula 1.

*A*˙ = argmax*pQ*(*A*|*Oads,Ochisq*) (1)

*A*

The probability *pQ*(*A*|*Oads,Ochisq*) is represented as a joint probability distribution function over a set of variables *A,Oads,Ochisq* incorporated into equation 2.

 (2)

After applying Bayes’s formula the joint probability in the nominator can be substituted with conditional probability as it is shown in equation 3. 

The denominator in formula is a constant value that does not depend on variable *A*. Moreover, we also have assumed both observations *Oads* and *Ochisq* are statistically independent. Therefore, the probability *pQ*(*A*|*Oads,Ochisq*) can be rewritten as equation 4

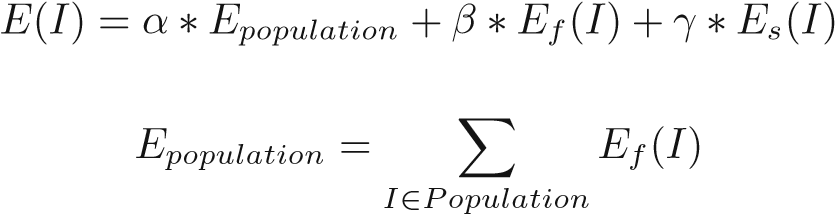
*pQ*(*A*|*Oads,Ochisq*) ∝ *pQ*(*Oads*|*A*)*pQ*(*Ochisq*|*A*)*PQ*(*A*) (4)

The methods for evaluating value of *Oads* and *Ochisq* for a given SQL query *Q* were described in section 3.1 and section 3.2 respectively.

## Extracting Signatures from SQL Queries

The details about the algorithm can be found in our previous work [2]. Therefore, to keep this paper self-contained here we only introduce the most important aspects.

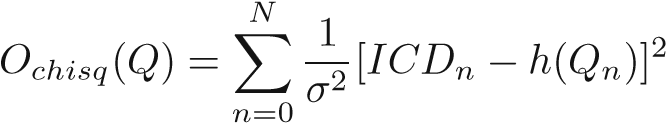
The proposed method exploits genetic algorithm, where the individuals in the population explore the log file that is generated by the SQL database. Each individual aims at delivering an generic rule (which is a regular expression in form "SELECT [a-z,]+ FROM patient WHERE name like [a-zA-z]+") that will describe visited log line. It is important for the algorithm to have an set of genuine SQL queries during the learning phase. The algorithm is divided into the following steps:

* Initialization. Each individual and line from log file is assigned. Each newly selected individual is compared to the previously selected in order to avoid duplicates.
* Adaptation phase. Each individual explores the fixed number of lines in the log file (the number is predefined and adjusted to obtain reasonable processing time of this phase).
* Fitness evaluation. Each individual fitness is evaluated. The global population fitness as well as rule level of specificity are taken into consideration, because we want to obtain set of rules that describe the lines in the log file.
* Cross over. Randomly selected two individuals are crossed over using algorithm for string alignment. If the newly created rule is too specific or too general it is dropped in order to keep low false positives and false negatives.

The fitness function, that is used to evaluate each individual, takes into account the particular regular expression effectiveness (number of times it matches to a given SQL query string), the level of specificity of such rule and the overall effectiveness of the whole population. The fitness function is described by equation 5, where *I* indicates the particular individual regular expression, *Epopulation* indicates the fitness of the whole population, *Ef* effectiveness of regular expression (number of times the rule fires), and *Es* indicates the level of specificity (in order to avoid too short regular expressions like ".\*"). The *α*, *β*, and *γ* are constants that normalize the overall score and weight each coefficient importance.

## Estimating Character Distribution for SQL Queries

The method is similar to the one proposed by C.Kruegel in [1]. The proposed character distribution model for describing the genuine traffic generated to web application. The Idealized Character Distribution (ICD) is obtained during the training phase from "clean" requests sent to web application. The IDC is calculated as mean value of all character distributions. During the detection phase the probability that the character distribution of a query is an actual sample drawn from its ICD is evaluated. For that purpose ChiSquare metric is used. The equation used for evaluating the value of variable *Ochisq*(*Q*) for a query *Q* is described by formula 7, where *N* indicates the length of a query *Q*, *ICD* the character distribution of all genuine SQL queries, *σ* the standard deviation form the *ICD*, and *h*()˙ the character distribution of a tested query *Q*.

 (7)

# Experiments and Results

In this section our evaluation methodology is described. The SQL Injection Attacks are conducted on php-based web service with state of the art tools for services penetration and SQL injection. The traffic generated by attacking tools are combined together with normal traffic (genuine queries) in order to estimate the effectiveness of the proposed methods. The genuine queries are both man-made and generated by web crawlers as well.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SNORT | ICD | SCALP | Proposed Methods |
| Effectiveness | 78,3% | 95,0% | 89,0% | 97,0% |

The web service used for penetration test is so called LAMP (Apache + MySQL + PHP) server with MySQL back-end. It is one of the most common worldwide used servers and therefore it was used for validation purposes. The server was deployed on Linux Ubuntu operation system. For penetration tests examples services developed in PHP scripts and shipped by default with the server are validated.

Attack injection methodology is based on the known SQL injection methods, namely: boolean-based blind, time-based blind, error-based, UNION query and stacked queries. For that purpose sqlmap tool is used. It is an open source penetration and testing tool that allows the user to automate the process of validating the tested services against the SQL injection flaws.

In order to avoid double-counting the same attack patterns during the evaluation process, we decided to gather first the malicious SQL queries generated by sqlmap (several hundreds of different injection trials). After that genuine traffic (generated by crawlers and during the normal web service usage) is gathered.

The proposed method for injection attack detection has been compared with known in the literature solutions, namely Apache SCALP, SNORT, and ICD. Apache SCALP is an analyzer of Apache server access log file. It is able to detect several types of attacks targeted on web application. The detection is a signature-based one. The signatures have form of regular expressions that are borrowed from PHP-IDS project.

The results show that SCALP and Snort are not efficient solutions for SQL attack detection. The advantage is that they can be easily fed with new signatures. However, most of the available rules are intended to detect very specific type of attacks that usually exploit very specific web-based application vulnerabilities.

Table 1. Effectiveness of injection attack detection

# Conclusions

In this paper we have extended our previous work presented in [2]. We have compared our method with known in the literature solutions for injection attack detection. Our experiments showed that available signatures-based solutions for SQL injection attacks are not efficient. Most of the available rules are intended to detect very specific type of attacks that usually exploit very specific web-based application vulnerabilities. The proposed algorithm combines advantages of ADS and signature-based methods. It allows the algorithm to achieve detection effectiveness that is significantly better when compared to signature-based methods only.

Acknowledgement. This work was partially supported by Applied Research Programme (PBS) of the National Centre for Research and Development (NCBR) funds allocated for the Research Project number PBS1/A3/14/2012 (SECOR)).

# References

1. Kruegel, C., Toth, T., Kirda, E.: Service specific anomaly detection for networkintrusion detection. In: Proc. of ACM Symposium on Applied Computing, pp. 201–208 (2002)
2. Choraś, M., Kozik, R., Puchalski, D., Hołubowicz, W.: Correlation Approachfor SQL Injection Attacks Detection. In: Herrero, Á., et al. (eds.) Int. Joint Conf. CISIS’12-ICEUTE’12-SOCO’12. AISC, vol. 189, pp. 177–185. Springer, Heidelberg (2013)
3. NSG, <http://www.ijcst.com/vol31/4/sridevi.pdf>
4. LESG, <http://www.cs.northwestern.edu/~ychen/Papers/LESG-ICNP07.pdf>
5. Shabtai, A., Menahem, E., Elovici, Y.: F-Sign: Automatic, Function-Based Signature Generation for Malware. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 41(4), 494–508 (2011)
6. Kong, D., Jhi, Y.-C., Gong, T., Zhu, S., Liu, P., Xi, H.: SAS: Semantics awaresignature generation for polymorphic worm detection. In: Jajodia, S., Zhou, J. (eds.) SecureComm 2010. LNICST, vol. 50, pp. 1–19. Springer, Heidelberg (2010)
7. SNORT. Project homepage, <http://www.snort.org/>
8. Sharma, M., Toshniwal, D.: Pre-clustering algorithm for anomaly detection andclustering that uses variable size buckets. In: 2012 1st International Conference on RecentAdvances in Information Technology (RAIT), pp. 515–519 (2012)
9. Adaniya, M.H.A.C., Lima, M.F., Rodrigues, J.J.P.C., Abrao, T., Proenca, M.L.:

Anomaly detection using DSNS and Firefly Harmonic Clustering Algorithm. In:

2012 IEEE International Conference on Communications (ICC), pp. 1183–1187 (2012)

1. Mazel, J., Casas, P., Labit, Y., Owezarski, P.: Sub-Space clustering, InterClustering Results Association and anomaly correlation for unsupervised network anomaly detection. In: 2011 7th International Conference on Network and Service Management (CNSM), pp. 1–8 (2011)
2. Yang, C., Deng, F., Yang, H.: An Unsupervised Anomaly Detection Approachusing Subtractive Clustering and Hidden Markov Model. In: Second International Conference on Communications and Networking in China, CHINACOM 2007, pp. 313–316 (2007)
3. Liang, H., Wei-wu, R., Fei, R.: An Adaptive Anomaly Detection Based onHierarchical Clustering. In: 2009 1st International Conference on Information Science and Engineering (ICISE), pp. 1626–1629 (2009)
4. Pons, P., Latapy, M.: Computing Communities in Large Networks Using Random Walks. Journal of Graph Algorithms and Applications 10(2), 191–218

(2006)

1. Liao, Q., Blaich, A., Van Bruggen, D., Striegel, A.: Managing networks throughcontext: Graph visualization and exploration. Computer Networks 54, 2809– 2824 (2010)
2. Ricciato, F., Fleischer, W.: Bottleneck Detection via Aggregate Rate Analysis: A Real Casein a 3G Network. In: Proc. IEEE/IFIP NOMS (2004)
3. Thottan, M., Ji, C.: Anomaly Detectionin IP Networks. IEEE Trans. Signal

Processing, Special Issue of Signal Processing in Networking 51(8), 2191–2204 (2003)

1. Rish, I., Brodie, M., Sheng, M., Odintsova, N., Beygelzimer, A., Grabarnik, G.,Hernandez, K.: Adaptive Diagnosis in Distributed Systems. IEEE Tran. Neural Networks 16(5), 1088–1109 (2005)
2. Gaddam, S.R., Phoha, V.V., Balagani, K.S.: K-Means+ID3: A Novel Methodfor Supervised Anomaly Detection by Cascading K-Means Clustering and ID3 DecisionTree Learning Methods. IEEE Transactions on Knowledge and Data Engineering 19(3), 345–354 (2007)
3. Ma, R., Liu, Y., Lin, X., Wang, Z.: Network Anomaly Detection Using RBFNeural Network with Hybrid QPSO. In: IEEE International Conference on Networking, Sensing and Control, ICNSC 2008, pp. 1284–1287 (2008)
4. Ma, J., Dai, G., Xu, Z.: Network Anomaly Detection Using Dissimilarity-BasedOne-Class SVM Classifier. In: International Conference on Parallel Processing Workshops, ICPPW 2009, pp. 409–414 (2009)
5. Soule, A., Salamatian, K., Taft, N.: CombiningFiltering and Statistical Methodsfor AnomalyDetection. In: Proc. of IMC Workshop (2005)