Vol.2012-MPS-89 No.10
2012/7/16

A Learning Algorithm of Threshold Value on the Automatic Detection of SQL Injection Attack

Daiki Koizumi,[†1] Takeshi Matsuda,[†1]
Michio Sonoda[†1] and Shigeichi Hirasawa[†2]

The SQL injection attack causes very serious problem to web applications which have database including personal data. To detect the SQL injection attack, the parsing and the black list based on the existed attack have been widely used. Those approaches, however, have some problems in terms of the size of list or calculation costs as the number of attacks increases. For this point, the authors have previously proposed a simple automatic detection algorithm of SQL injection attack. This algorithm requires to calculate the contained rate of suspicious characters with input sequence. This rate would be compared with a known real-valued threshold. This paper proposes the learning algorithm to choose the real-valued threshold from training data sets. Furthermore, some criteria would be considered and their performances would also be examined.

1. Introduction

The SQL injection attack was firstly reported in 1999[2]. It consists of insertion or "injection" of a SQL query via the input data from the client to the application[3]. It causes very serious problem to web applications that involve database including personal data. To detect the SQL injection attack, the parsing and the blacklist built based on the existed attack have been widely used. However, the conventional methods based on blacklist built approaches have confronted at least two problems. The first is the huge cost of updating large blacklist built, and the second is the calculation cost for automatic attack detection with large blacklist built.

For these two problems, the authors have already proposed the online algorithm for automatic detection of SQL injection attack[4]. This algorithm requires to calculate the contained rate of suspicious characters with input sequence. This rate is compared with a known real-valued threshold. In the authors' previous research[4], this threshold was empirically determined as a known constant. However, in general, this threshold is an unknown value and should be learned from a given training set of data. This paper proposes a learning algorithm to choose a real-valued threshold. Furthermore, some criteria will considered and their performances will also be examined.

The rest of this paper is organized as the followings. Section 2 gives some definitions and automatic detection algorithm of SQL injection attack which was previously proposed where its threshold value is defined as a known constant. Section 3 proposes the learning algorithm of the threshold value with some formulations. Section 4 shows some leaning examples with artificial data. Section 5 gives their discussions. Finally, Section 6 concludes this paper.

2. The Automatic Detection Algorithm of SQL Injection Attack[4]

2.1 Preliminary

Suppose that l is an input sequence through web application to the SQL database. Note that each input l has a label either attack or normal. The purpose of automatic SQL injection attack detection is to estimate the label of l correctly. Our proposed algorithm requires some known finite characters $s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{iii}}, \cdots$. These are suspicious characters contained in SQL injection attack inputs such as space, semicolon, single-quotation, left and right round brackets, and so on. Furthermore, let S_k, $k = 1, 2, \cdots, m$ be power sets of known characters $s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{iii}}, \cdots$ where they are defined as sets of characters except for the empty set. If two characters $s_{\mathbf{i}}$ and s_{ii} are defined as space and semicolon, then their three power sets of known characters can be $S_1 = \{s_{\mathbf{i}}\}, S_2 = \{s_{\mathrm{ii}}\}$, and $S_3 = \{s_{\mathbf{i}}, s_{\mathrm{ii}}\}$.

2.2 The Automatic Detection Algorithm

With the above definitions, the automatic detection algorithm was proposed[4]. Furthermore, the theoretical performance was analyzed in terms of statistical prediction problem[1]. The following is the brief description of the automatic detection algorithm.

(1) Setting up known values

†1 Faculty of Information Technology and Business, Cyber University
†2 Research Institute for Science and Engineering, Waseda University

(a) Choose a set of characters S_k.
(b) Set a threshold as a real value $\alpha \in [0,1]$.

(2) Calculating the content rate of suspicious characters $x_{k,l}$ which is defined as,

$$x_{k,l} = \frac{\#S_k}{|\,l\,|}\,, \tag{1}$$

where $\#S_k$ denotes the size of S_k, and $|\,l\,|$ denotes the length of the input sequence.

(3) Automatic detection
Determine each input's label (normal or attack input) by the following function $d\,(x_{k,l}, \alpha)$:

$$d\,(x_{k,l}, \alpha) = \begin{cases} 0 & if\ \ x_{k,l} \le \alpha; \\ 1 & if\ \ x_{k,l} > \alpha. \end{cases} \tag{2}$$

Eq. (2) means that if detection result is normal, then its value is zero, otherwise the result is attack and its value is one.

Example 2.1 (Automatic Detection of Attack Input)
Let l = "DROP sampletable;--" be an attack input where its length $|\,l\,| = 19$. Suppose that a character set S_{13} contains space, semicolon, and left round brackets. Furthermore, the threshold value α is set to 0.10.

Since the input l contains one space and one semicolon among characters in S_{13}, a numerator in Eq. (1) becomes $\#S_{13} = 2$. Therefore, according to Eq. (1),

$$\begin{aligned} x_{13,\,l} &= \frac{2}{19} \\ &= 0.1052\cdots \end{aligned}$$

With the above $x_{13,\,l}$, the detection result by Eq. (2) becomes the following:

$$\begin{aligned} d\,(x_{13,\,l}, \alpha) &= d\,(0.1052, 0.10) \\ &= 1\,. \end{aligned}$$

Thus l is detected as an attack input.

□

2.3 Performance Evaluation with Artificial Data

For evaluation of the above algorithm, the artificial data was composed[4]. Those data cover the typical types of SQL injection attack input as well as normal input among common web forms. The number of types of attack inputs was 624, on the other hand, that of normal inputs was 234. Those data were converted to the fields of single and multibytes characters, wiki, emoticon etc. These types were assumed to be input as IDs, passwords, names, and addresses etc.[4].

If the real operating situation is considered, the label of each input (either normal or attack) is unknown. Therefore, the mixture data of both labels were used for simulations in evaluations[4]. Let $0 \le P_N \le 1$ be the correct detection rate for normal input and let $0 \le P_A \le 1$ be the correct detection rate for attack input. Furthermore, if the ratio of the number of the normal input against attack input is $0 \le \beta \le 1$, the total detecting rate $0 \le \mu \le 1$ can be calculated by the following:

$$\mu\,(S_k, \alpha, \beta) = \beta P_N + (1-\beta)\,P_A\,. \tag{3}$$

For evaluation, the value of α was empirically chosen as a constant. On the other hand, various values of β were taken with its interval $0 \le \beta \le 1$. For the remained S_k, the following objective function was assumed to chose the optimal character set of S^*.

$$S^* = \arg\max_{S_k} \left[\mu\,(S_k, \alpha, \beta)\right]\,. \tag{4}$$

With the above Eq. (4), the sensitive analysis was considered for discussions[4].

3. The Proposed Learning Algorithm of Threshold Value

3.1 Formulation and Criterion

In general, the threshold value is unknown and should be learned from real observed data. If S^* has been already determined and the candidates of α_j, $j = 1, 2, \cdots, N$ have been obtained, then the following can be defined as similar form of Eq. (4).

$$\alpha^* = \arg\max_{\alpha_j} \left[\mu\,(S_k, \alpha_j, \beta)\right]\,. \tag{5}$$

Since the label of input l is unknown at the real operation, the value of β, which is the weight of normal input against attack input is also unknown. Therefore, Eq. (4) and (5) can be achieved with several criteria. One of them is that assuming

the probability distribution of $p(\beta)$ to take the expectation of $\mu(S_k, \alpha_j, \beta)$ with respect to β. For numerical approximation, suppose $\beta_m, m = 1, 2, \cdots, M$ is sampled on the interval $0 \leq \beta \leq 1$. Then, such criterion can be formulated as the following:

$$\{\alpha^{**}, S^{**}\} = \arg\max_{\alpha_j} \max_{S_k} \left[\sum_{m=1}^{M} p(\beta_m)\ \mu(S_k, \alpha_j, \beta_m) \right]. \tag{6}$$

Note that α^{**}, S^{**} maximize the expected total detecting rate $\mu(S_k, \alpha_j)$ in Eq. (6).

For the other criteria, both the expected total detecting rate and the absolute value of slope of regression line can be considered which is more restrictive. This criterion also takes into account the stability of $\mu(S_k, \alpha_j)$ with respect to β.

$$\begin{aligned} &\{\alpha^{***}, S^{***}\} \\ &= \arg\max_{\alpha_j} \max_{S_k} \left[\sum_{m=1}^{M} p(\beta_m)\ \mu(S_k, \alpha_j, \beta_m) \right. \\ &\quad \left. - \left| \frac{M \sum_{m=1}^{M} \beta_m \mu(S_k, \alpha_j, \beta_m) - \left(\sum_{m=1}^{M} \beta_m \right) \left(\sum_{m=1}^{M} \mu(S_k, \alpha_j, \beta_m) \right)}{M \sum_{m=1}^{M} (\beta_m)^2 - \left(\sum_{m=1}^{M} \beta_m \right)^2} \right| \right]. \end{aligned} \tag{7}$$

Note that α^{***}, S^{***} maximize the sum of the expected total detecting rate and the slope of regression line. In Eq. (7), the second term on the right hand side expresses the absolute value of slope of the regression line $\mu(S_k, \alpha_j)$ where β is its domain.

3.2 The Proposed Algorithm

With training data set that contain pairs of input sequence l and its label, the following learning algorithm of threshold value α^{**} or α^{***} would be proposed.

(1) With various candidate pairs of S_k and α_j, execute automatic detection algorithm with Eq. (1) and (2).
(2) Calculate the total detecting rate $\mu(S_k, \alpha_j, \beta)$ with P_N, P_A, and $0 \leq \beta \leq 1$.
(3) Taking β_m for numerical approximation to choose the optimal set of S^{**} and α^{**} by Eq. (6)(or S^{***} and α^{***} by Eq. (7)).

4. Evaluations of the Proposed Algorithm

4.1 Conditions

For evaluations, the artificial data mentioned in subsection 2.3 was used. The number of types of attack inputs is 624, those of normal inputs is 234 where the data cover the single and multibytes characters, wiki, emocticon etc. Note that the data was assumed IDs, passwords, names, and addresses etc.

For known finite characters, the five characters were chosen as Table 1. These characters are same as our previous simulations[4].

Table 1 Known Characters

Name	Character
$s_{\mathbf{i}}$	Space
s_{ii}	Semicolon (;)
s_{iii}	Single Quotation (')
s_{iv}	Right Parenthesis ())
s_{v}	Left Parenthesis(()

With the above five characters, the following twenty six power sets can be defined as,

$S_1 = \{s_{\mathbf{i}}, s_{\mathrm{ii}}\}, S_2 = \{s_{\mathbf{i}}, s_{\mathrm{iii}}\}, S_3 = \{s_{\mathbf{i}}, s_{\mathrm{iv}}\}, S_4 = \{s_{\mathbf{i}}, s_{\mathrm{v}}\}, S_5 = \{s_{\mathrm{ii}}, s_{\mathrm{iii}}\},$
$S_6 = \{s_{\mathrm{ii}}, s_{\mathrm{iv}}\}, S_7 = \{s_{\mathrm{ii}}, s_{\mathrm{v}}\}, S_8 = \{s_{\mathrm{iii}}, s_{\mathrm{iv}}\}, S_9 = \{s_{\mathrm{iii}}, s_{\mathrm{v}}\}, S_{10} = \{s_{\mathrm{iv}}, s_{\mathrm{v}}\},$
$S_{11} = \{s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{iii}}\}, S_{12} = \{s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{iv}}\}, S_{13} = \{s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{v}}\}, S_{14} = \{s_{\mathbf{i}}, s_{\mathrm{iii}}, s_{\mathrm{iv}}\},$
$S_{15} = \{s_{\mathbf{i}}, s_{\mathrm{iii}}, s_{\mathrm{v}}\}, S_{16} = \{s_{\mathbf{i}}, s_{\mathrm{iv}}, s_{\mathrm{v}}\}, S_{17} = \{s_{\mathrm{ii}}, s_{\mathrm{iii}}, s_{\mathrm{iv}}\}, S_{18} = \{s_{\mathrm{ii}}, s_{\mathrm{iii}}, s_{\mathrm{v}}\},$
$S_{19} = \{s_{\mathrm{ii}}, s_{\mathrm{iv}}, s_{\mathrm{v}}\}, S_{20} = \{s_{\mathrm{iii}}, s_{\mathrm{iv}}, s_{\mathrm{v}}\},$
$S_{21} = \{s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{iii}}, s_{\mathrm{iv}}\}, S_{22} = \{s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{iii}}, s_{\mathrm{v}}\}, S_{23} = \{s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{iv}}, s_{\mathrm{v}}\},$
$S_{24} = \{s_{\mathbf{i}}, s_{\mathrm{iii}}, s_{\mathrm{iv}}, s_{\mathrm{v}}\}, S_{25} = \{s_{\mathrm{ii}}, s_{\mathrm{iii}}, s_{\mathrm{iv}}, s_{\mathrm{v}}\},$
$S_{26} = \{s_{\mathbf{i}}, s_{\mathrm{ii}}, s_{\mathrm{iii}}, s_{\mathrm{iv}}, s_{\mathrm{v}}\}.$

4.2 Simulations

(1) Simulation 1
Choose five pairs of $\{\alpha^{**}, S^{**}\}$ in descending order according to the criteria in Eq. (6).
(2) Simulation 2
Choose five pairs of $\{\alpha^{***}, S^{***}\}$ in descending order according to the cri-

teria in Eq. (7).

4.3 Results

Table 2 and 3 were obtained for Simulation 1 and 2, respectively.

Table 2 Result of Simulation 1

Rank	S^{**}, α^{**}	Value in Eq. (6)
1st	$S_{22}, \alpha = 0.08$	0.9170
2nd	$S_{12}, \alpha = 0.08$	0.9154
3rd	$S_{22}, \alpha = 0.09$	0.9147
4th	$S_{21}, \alpha = 0.08$	0.9142
5th	$S_{14}, \alpha = 0.02$	0.9135

Table 3 Result of Simulation 2

Rank	S^{***}, α^{***}	Value in Eq. (7)
1st	$S_{12}, \alpha = 0.09$	0.9032
2nd	$S_{12}, \alpha = 0.08$	0.8994
3rd	$S_{22}, \alpha = 0.11$	0.8930
4th	$S_{23}, \alpha = 0.10$	0.8903
5th	$S_{21}, \alpha = 0.11$	0.8879

5. Discussions

From Table 2 in Simulation 1, we can see that the pair S_{22} and $\alpha = 0.08$ maximizes the expected total detecting rate $\mu(S_k, \alpha_j)$ in Eq. (6). In our previous research[4], the pair $S_{12}, \alpha = 0.08$ was empirically chosen. According to Table 2, relatively superior pair was discovered with the criterion in Eq. (6). Figure 1 shows the plot of the top three pairs of S^{**} and α^{**} where the vertical axis is the value of μ and the horizontal axis is $0 \leq \beta \leq 1$. According to Eq. (6), the superior μ of the pair S_{22} and $\alpha = 0.08$ can be observed comparing to the empirically chosen pair S_{12} and $\alpha = 0.08$ in Figure 1. In Figure 1, the pair S_{12} and $\alpha = 0.08$ gives the most flattest line among three lines, however, the other two pairs give the relatively superior values according to Eq. (6).

From Table 3 in Simulation 2, S_{12} and $\alpha = 0.09$ are obtained as the optimal pair according to Eq. (7). The second is the pair of S_{12} and $\alpha = 0.08$. Figure 2 shows the same sort of plot as previous Figure 1. Since Eq. (7) emphasizes the absolute value of the slope, more flatter line can be chosen. In this criterion, S_{12}, which contains three characters, was superior to the others, whereas S_{22} was superior to them in the criteria Eq. (6).

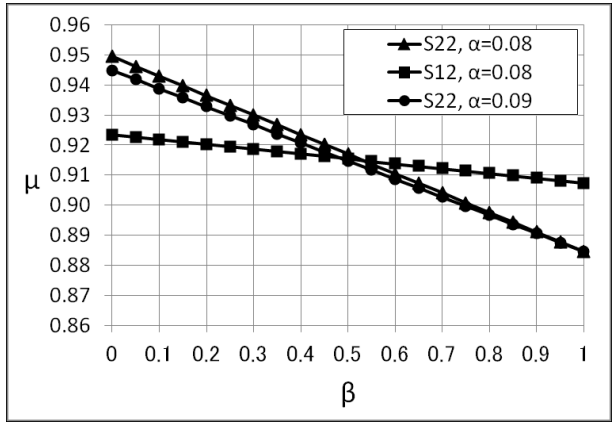

Fig. 1 Learning Results based on Eq. (6)

Figure 3 shows the effect of various values of α in S_{12}. From Figure 3, the more the value of α increases, the more larger the value of slope of the line becomes. Since Eq. (7) gives the penalty of the larger value of the slope, $\alpha = 0.09$ is more likely to be chosen.

Figure 4 shows the effect of various values of α in S_{21}. Figure 4 also shows that the more the value of α increases, the more the value of slope of the line becomes larger. But the increasing degree of the slope in S_{21} is relatively larger than that of S_{12}. This result can be interpreted as the effect of the character of Right Parenthesis which is the only contained character in S_{21}.

Figure 5 shows the effect of various sets among S_{12}, S_{21}, and S_{24} where those thresholds are the constant $\alpha = 0.08$. From Figure 5, the detecting performances of S_{12} and S_{21} are similar, however, that of S_{24} is the relatively poor. It can be observed that S_{24} is the only set which does not contain Semicolon.

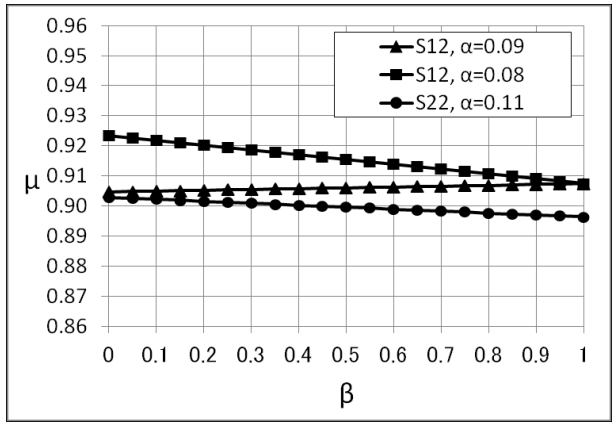

Fig. 2 Learning Results based on Eq. (7)

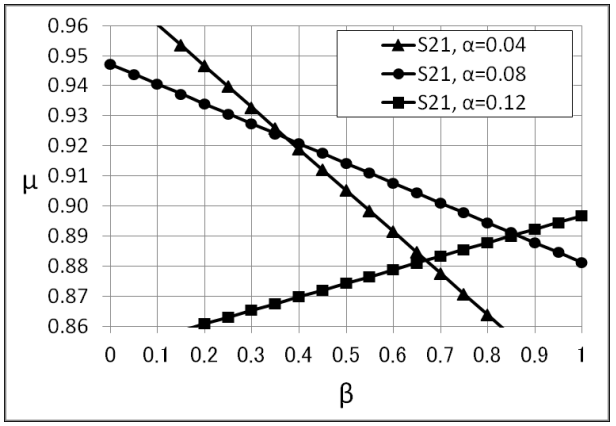

Fig. 4 Detecting Performance of S_{21} with Various Thresholds

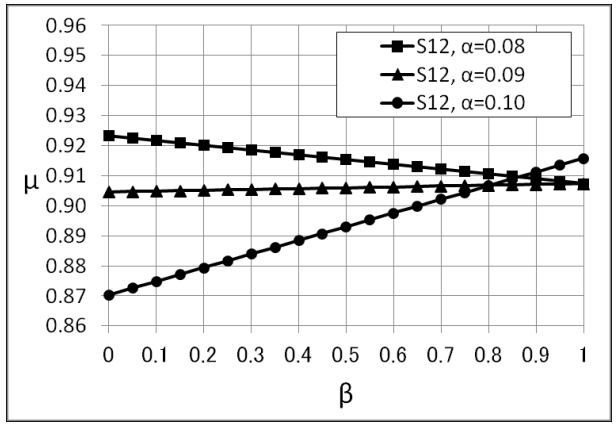

Fig. 3 Detecting Performance of S_{12} with Various Thresholds

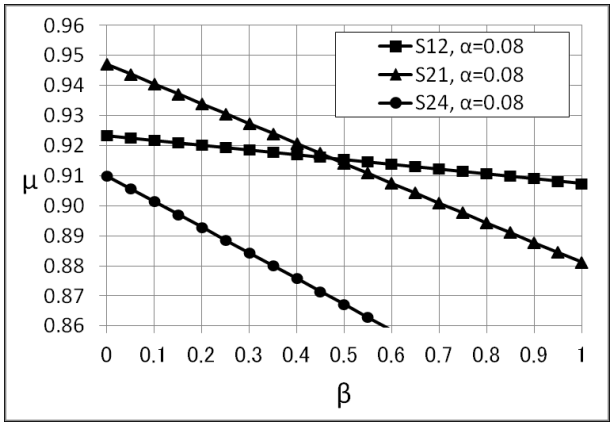

Fig. 5 Detecting Performance of S_{12}, S_{21}, and S_{24} with a Constant Thresholds ($\alpha = 0.08$)

6. Conclusions

This paper proposed the learning algorithm to choose the real-valued threshold for automatic detection of SQL injection attack. Furthermore, some learning criteria were considered and their performances were also examined with artificial data. As a result, the certain effectiveness was observed with the proposed algorithm and thus seeking the unknown threshold value can be possible with training sets of data.

For future research, predictive performance should be examined with unknown data sets. Furthermore, the detecting performance with the real SQL injection data should also be considered.

Acknowledgments

This research is partially supported by the Grant-in-Aid for Scientific Research (C) No.23501178 of the Japan Society for the Promotion of Science (JSPS).

This research is partially supported by the Grant-in-Aid for Scientific Research (B) No.23740094 of the Japan Society for the Promotion of Science (JSPS).

This research is partially supported by the 44uh Kurata Grants of the Kurata Memorial Hitachi Science and Technology Foundation.

References

1) Takeshi Matsuda, Daiki Koizumi, Michio Sonoda, and Shigeichi Hirasawa, "On Predictive Errors of SQL Injection Attack Detection by the Feature of the Single Character," Proceeding of 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC2011), pp.1722–1727, Oct.2011.
2) NT Web Technology Vulnerabilities [Online], `http://www.phrack.com/issues.html?issue=54`
3) The Open Web Application Security Project (OWASP), SQL Injection [Online], `https://www.owasp.org/index.php/SQL_Injection`
4) Michio Sonoda, Takeshi Matsuda, Daiki Koizumi, and Shigeichi Hirasawa, "On Automatic Detection of SQL Injection Attacks by the Feature Extraction of the Single Character," Proceedings of the 4th International Conference on Security of Information and Networks (SIN2011), pp.81–86, Nov.2011.