SQL Injection Attack Detection Method Using Structure Equation Models and Visualization of Data

By Matsuda. T.
Shizuoka Institute of Science and Technology, Shizuoka, Japan

Abstract

This paper proposed a new detection method of SQL injection attack based on the structure equation model and visualized the data of the SQL injection attack.

1. Introduction

SQL injection attack is very simple web application attack, and it is used to get an illegality access into the database of web applications. SQL injection attack is a character strings that includes malicious SQL grammar. So, it is very important to check the strings which are inputted into web applications, and the basic method for preventing is to escape some symbols of the input strings. To use the prepared statement is more valid measure to prevent web application. Many prevention and detection methods of web application attacks had been developed. However, the Open Web Application Security Project (OWASP) warns this vulnerability in OWASP Top 10 even now. Therefore, it is important to develop more effective detection measure of the SQL injection attack.

In this study, we proposed a detection method of SQL injection attack using the structure equation model. We treat the label data of attack and normal as a hidden variables of the structure equation model and calculate the quantity of the attack and the normal features by using the proposed model for visualizing the data of SQL injection attacks. We compared the detection precision between our proposed method and ModSecurity to evaluate the proposed method.

2. SQL Injection Attack

In this section, we will give the brief introduction of SQL injection attack. Web application receives the input data from web browser, and sends the data into the database. Therefore, if some malicious codes are including in the input data, an arbitrary command of SQL is executed in the database of the web applications. For example, assume that a user can display user's page on the website of "http://www.sqlinjection.com/" (unreal website) by inputting the ID "777" in the URL input field. Then, the web application makes the following URL parameter.

http://www.sqlinjection.com/?id=777

If a measure of this application is incomplete, the following input data becomes SQL injection attack.

777' or 'XYZ '= ' XYZ

If the query

http://www.sqlinjection.com?id=777' or 'XYZ '= ' XYZ

is executed, there is a fear that another all information of database may leak, because XYZ = XYZ is always true. This attack has the symbol of the single quote, and it is very important to make the attack succeed. If single quote is replaced with double quote, then this attack is deauthorized. The more basic measure is to use the technique of the prepared statement. However, for example, since there is the possibility of the mistake at the time of development, it is not easy to take preventive measures against SQL injection attacks. So, we considered that the detection method of SQL injection attack to prevent web applications that repair is difficult.

3. Proposed Method

The purpose of this study is to develop a detection method using mathematical method. To analyze the structure of SQL injection attack data, we prepared 2779 attack sample and 444 normal sample. We mainly collected attack sample from website of OWASP, and our attack sample includes a real attack data. On the other hand, we collected normal sample from website, but we generated a lot of normal sample. Normal sample includes name, id, password, wiki grammar and emoticon. Therefore, our normal sample has a lot of symbols. Before we introduce our proposed method, we explain the property of the distribution of SQL injection attack strings.

In the previous our study (Matsuda 2013), the distribution of SQL injection attack strings (Fig. 1) may be approximated by a zeta distribution. Zeta distribution is defined by

$$p(x|a) = \frac{1}{\zeta(a)x^{-a}}.$$

Here, $\zeta\left(a\right)=\sum_{x=1}^{\infty}\frac{1}{x^{-a}}$ is Riemann zeta function.

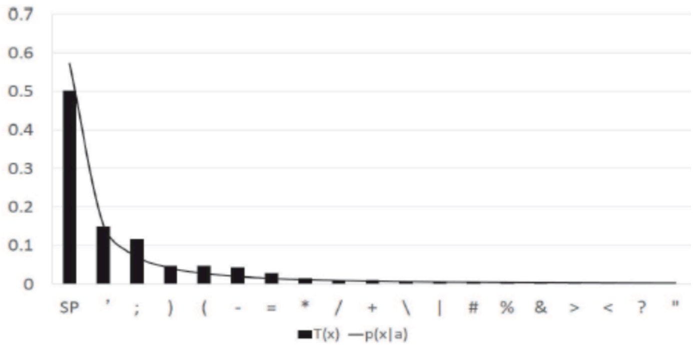


FIGURE 1. The distribution of SQL injection attack strings

The curve of Fig 1 is the zeta distribution approximating the distribution of SQL injection attacks strings. The horizontal axis indicates symbols in the set of attack strings, and the vertical axis indicates the content rate of the corresponding symbols. On the other hand, Fig 2 shows the distribution of normal strings. We can see that the zeta distribution is not fit well with the distribution of normal strings.

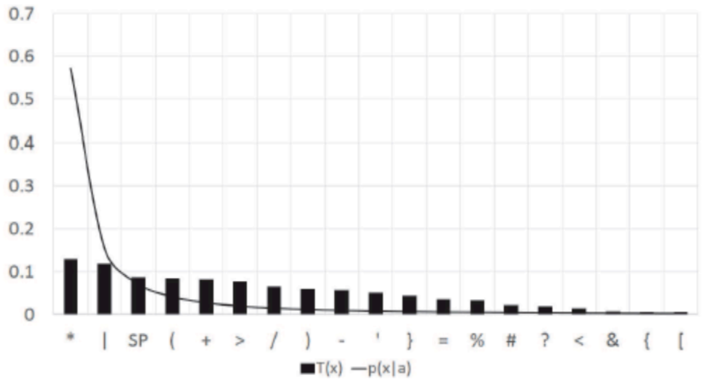


FIGURE 2. The distribution of normal strings

Note that the horizontal axis of the distribution of normal strings is different from the one of the distribution of SQL injection attacks. This means that the appearance symbols between the attack and the normal are differences. In this study, we considered that the feature extraction method of the attack and normal by using the structure equation model which has a latent variables.

In this study, we proposed a simple model that imitates the feature of Fig. 1.

$$\begin{aligned} y_x &= a_0 + a_1 x + a_2 x^2 + \varepsilon_1 \\ a_0 &= b_{00} + b_{10} t + \varepsilon_2 \\ a_1 &= b_{01} + b_{11} t + \varepsilon_3 \\ a_2 &= b_{02} + b_{12} t + \varepsilon_4 \end{aligned}$$

Here, $\varepsilon_i \sim N\left(0,\sigma_i^2\right)$ and $N\left(0,\sigma_i^2\right)$ shows the normal distribution with the mean 0 and the variance σ_i^2. The variable x is corresponding to the symbols in the following way.

$$\begin{aligned} x=1 &: \text{Space} \\ x=2 &: \text{‘} \\ x=3 &: ; \\ x=4 &:) \\ x=5 &: (\end{aligned}$$

The variable y_x is computed from

y_x = (the number of x on the attack sample set) / (the total number of symbols on the attack sample set).

The variable $t \in \{0,1\}$ is the label data of the attack or normal. We define $t=1$ (resp. $t=0$) as an attack (resp. a normal). Next, we summarize the parameter estimation method of the proposed model.

Firstly, we calculate the parameter a_0, a_1, a_2 from sample $\left\{\left(x_i, y_{x_i}\right)\right\}_{i=1}^{I}$ by using the maximum likelihood method. Concretely, we compute $a_0^{(i)}, a_1^{(i)}, a_2^{(i)}$ which maximize the following likelihood function.

$$\prod_{x_i=1}^{5} \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left[-\frac{\left(y_x-\left(a_0+a_1x_i+a_2x_i^2\right)\right)^2}{2\sigma_1^2}\right].$$

Finally, we calculate the parameter $b_{00}, b_{10}, b_{01}, b_{11}, b_{02}, b_{12}$ from $\left\{\left(a_0^{(i)}, a_1^{(i)}, a_2^{(i)}, t_i\right)\right\}_{i=1}^{I}$, and the following results are obtained.

$$\begin{aligned} b_{00} &= 0.367642642642704 \\ b_{10} &= 0.520581649276325 \\ b_{01} &= 0.0093468468468900 \\ b_{11} &= -0.427524704823484 \\ b_{02} &= -0.00379129129129632 \\ b_{12} &= 0.061190887111532 \end{aligned}$$

The above parameter estimation result is obtained from our prepared sample data (2779 attacks and 444 normals). From the above discussion, the feature of attack sample ($t=1$) is expressed by

$$y = 0.8882 - 0.4182x + 0.0574x^2.$$

Similarly, the feature of normal sample ($t=0$) is expressed by

$$y = 0.3676 + 0.0093x - 0.038x^2.$$

Fig. 3 shows that the feature of the attack sample and the normal sample. The red curve indicates the feature of SQL injection attack. We can consider that the coordinate $(1, 0.5274)$ shows the attack feature value on the symbol of Space. Similarly, the attack feature values of single quote, semi colon, right parenthesis and left parenthesis are 0.2815, 0.1503, 0.1339 and 0.2323, respectively. The normal feature values are obtained from the blue curve in Fig 3. By using the attack and normal feature values, we can detect SQL injection attacks. The detection algorithm is very simple. We transform a target data to the coordinates, and compute the residual error between those coordinates and the feature curves. If the residual value with the attack feature curve is smaller than the normal one, we judge the target data as an attack. Moreover, we can visualize the feature of SQL injection attack data using those feature values. Specifically, we corresponded the attack feature value to R-value (0 to 255), and the normal feature value to B-value (0 to 255). We introduce the above details in the next section.

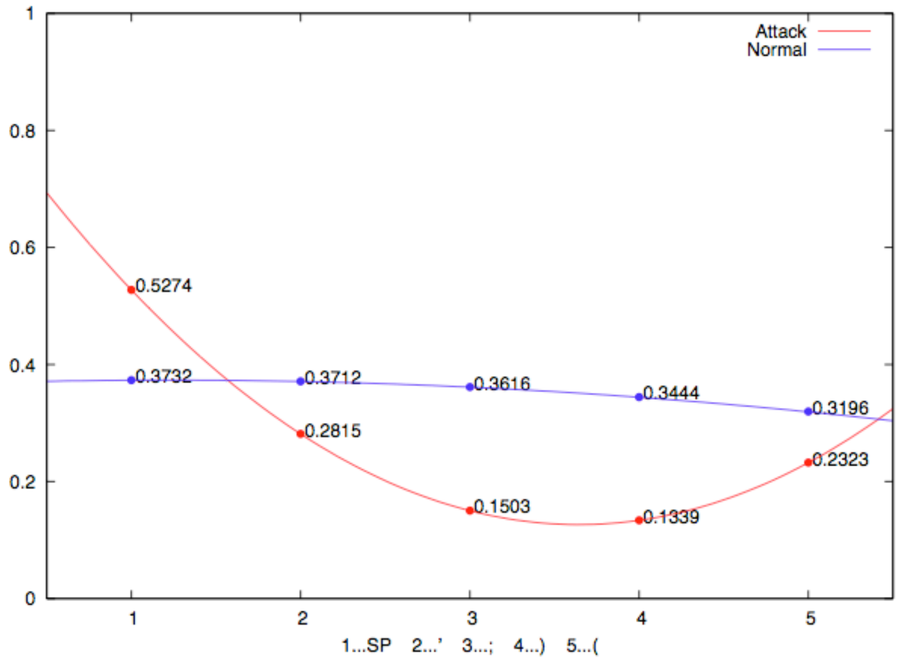


FIGURE 3. Features of the attack and the normal

4. Detection Method and Visualization

We compute the residual to detect SQL injection attack, and we check which curve is near the given new coordinates. To evaluate our proposed detection method, we prepared 200 attacks and 50 normals as the test data. Then, our proposed method detected 96% of attack data and 100% of normal data collectedly. On the other hand, ModSecurity detected 100% of attack data. However, ModSecurity misdetected 70% of normal data. For example, ModSecurity did not detect normal data including some symbol such as

https://www.google.co.jp/?gws rd=ssl#q=
(%EF%BE%9F%E2%88%87%5E*)%EF%BD%B5 %EF%BE%8A%EF%BE%96%E2%99%AA .

Our proposed method calculates the quantity of the feature of the attack and normal. Therefore, we can make some colors correspond to these quantities. The values of Table 1 are calculated from the following linear transformation.

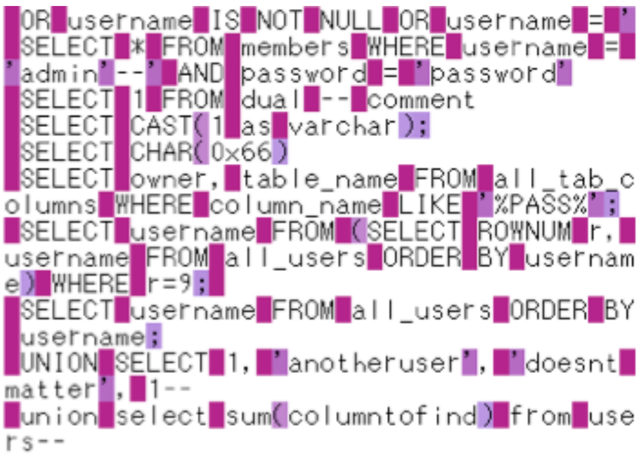
$$[0,1] \rightarrow [0,255]$$

The domain $[0,1]$ indicates the attack and normal feature value, and the range $[0,255]$ indicates the color strength of Red and Blue. For example, we can color the data of SQL injection attack as the following Fig 4. The attack features and normal features are colored with red and blue, respectively.

TABLE 1. The coloring of the attack feature and the normal featreu

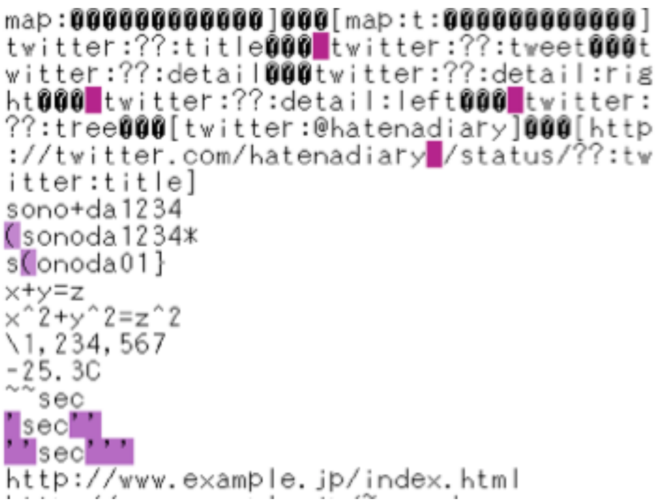
Symbol	Attack feature	Normal feature
Space	95.165541	134.498738
‘	94.648649	71.774075
;	92.198198	38.323205
)	87.814189	34.146130
(	81.496622	59.242849

Fig 4 is the data of SQL injection attack. We can see that many symbols are colored by red. On the other hand, Fig 5 is the data of normal. In comparison with the data of SQL injection attack, we can see that the number of symbols colored with red is small. By using this technique, we may make some teaching materials about SQL injection attack.



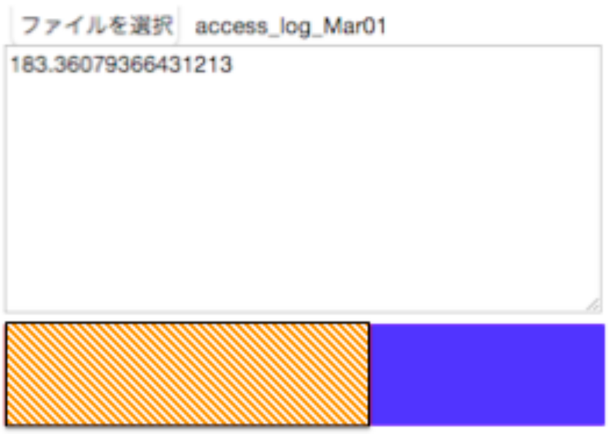
```
 OR username IS NOT NULL OR username = '
 SELECT * FROM members WHERE username =
'admin'--' AND password = 'password'
 SELECT 1 FROM dual -- comment
 SELECT CAST( 1 as varchar );
 SELECT CHAR( 0x66 )
 SELECT owner, table_name FROM all_tab_c
olumns WHERE column_name LIKE '%PASS%';
 SELECT username FROM (SELECT ROWNUM r,
username FROM all_users ORDER BY usernam
e) WHERE r=9;
 SELECT username FROM all_users ORDER BY
username;
 UNION SELECT 1, 'anotheruser', 'doesnt
matter', 1--
 union select sum(columntofind) from use
rs--
```

FIGURE 4.   Coloring of the SQL injection attack



```
map:000000000000]000[map:t:000000000000]
twitter:??:title000 twitter:??:tweet000t
witter:??:detail000twitter:??:detail:rig
ht000 twitter:??:detail:left000 twitter:
??:tree000[twitter:@hatenadiary]000[http
://twitter.com/hatenadiary /status/??:tw
itter:title]
sono+da1234
(sonoda1234*
s(onoda01}
x+y=z
x^2+y^2=z^2
\1,234,567
-25.3C
~~sec
'sec''
''sec'''
http://www.example.jp/index.html
```

FIGURE 5.   Coloring of the normal data



183.36079366431213

FIGURE 6.   Visualization of the log data of Apache

The coloring of data can be used for another purpose. For example, it is hard to find the trace of cyber attack from huge log data. However, we can also visualize the log data like figure 6 using our proposed model in the similar way. To validate our proposed method using real log data is our important future work.

5. Conclusion

In this study, we proposed the detection method of SQL injection attack using the structure equation model. The character of our proposed model can treat SQL injection attack as a function. Our future work is to investigate the property of these functions and to evaluate the visualizing method on the real web service.

REFERENCES

The Open Web Application Security Project (OWASP), https://www.owasp.org/index.php/Main Page

ModSecurity, https://www.modsecurity.org

E. H. Cheon, Z. Huang and Y. S. Lee, "Preventing SQL Injection Attack Based on Machine Learn- ing," International Journal of Advancements in Computing Technology(IJACT) Volume5, Num- ber9, May 2013.

T. Matsuda, "Feature extraction of Web appli- cation attacks based on zeta distributions," 2013 World Congress on Internet Security (WorldCIS), pp.119-121, 2013.