**SQL Injection Attack Detection Framework Based on HTTP Traffic**

4 modules: data collection, data cleaning, feature representation and model building.

The framework can detect SQL injection attacks in URL, POST and HTTP header rows.

Howard et al. [2] proposed an automatic signature generation

system for SQL injection attacks based on machine learning. The

system first actively collected 30000 SQL injection attack traffic, con-

structed statistical features by using SQL reserved words, firewall

rules and expert knowledge, and then used hierarchical clustering

algorithm to cluster the samples and features successively. They

derived the characteristics of each cluster, and trained the logistic

regression model as a signature for each cluster. Compared with the

rule-based intrusion detection system (Snort and Bro) and the Web

application firewall (ModSec), the system shows strong competi-

tiveness in the detection effect of SQL injection.

Kar et al. [3] proposed a detection method for SQL injection attacks

deployed in database firewalls. By converting SQL statements into

token sequences that maintain their syntactic structure, a graph

with tokens as nodes is generated, the interaction between tokens

is used as a weighted edge, and then the central measure of nodes

is used as a feature to train the SVM classifier. A variety of different

parameter settings are experimented in detail. The accuracy of the

final model is 99.47%. The false positive rate was 0.31%.

**Analysis**

Four characteristics of SQL injection attack traffic: Firstly, SQL injection conforms to SQL grammar rules and must contain SQL reserved words or built-in functions, etc. Secondly, annotation sym-

bols in SQL injection are short and composed of 1 or 2 characters, but they are very important features. Thirdly, SQL injection statements may appear anywhere in the traffic. If the traffic is truncated, information will be lost. Last, strings or numbers that appear in the SQL injection are artificial random inputs that have no meaning for the detection system.

**Workflow**

The detection framework has two phases: the training phase and the prediction phase. The training phase consists of four modules, and finally a detection model is obtained. The prediction phase uses the detection model to predict the label of the input flow.

**Data cleaning**

1.The decoding. In the decoding operation, if the HTTP method is GET, the URL needs to be decoded first. The parameter string in the URL is made up of key-value pairs of the form Key = Value, with an & interval between each pair. The Payload is decoded in URL, HTML and Base64, which can effectively help detection model learn the characteristics of SQL injection attack.

2.Uniform case. A change in the case of the characters in the tag is perceived by the computer as two different tags, which greatly increases the dimension of lexical characteristics. Second, the case of the characters in the traffic has no meaning for the detection task. At the same time, some attackers use case bypass, such as select rewritten to SELECT, in order to avoid the intrusion detection system inspection. Therefore, in this paper, characters in Payload are uniformly converted to lowercase letters in preprocessing.

3.Replace. All reserved words, functions, etc. in SQL language are composed of ASCII-encoded characters. In order to reduce the interference of irrelevant information, non-ASCII-encoded strings are replaced with the fixed unk tag. Substitution processing is adopted in the detection framework instead of filtering processing. Filtering will result in loss of information leading to omissions, and may lead to connection of unrelated parts leading to false positives. The ideal way to handle this is to replace a combination of characters of the same type with the same tag.

4.Traffic standardization. The string or number that appears in

the Payload is an artificial random input that is not meaningful for the detection task. In this paper, Payload is standardized in the data cleaning module. For different types of artificial input, different identifiers are defined to replace the traffic so as to reduce the interference of irrelevant information in the training of the model. The number in Payload is replaced with ’num’ and the string is replaced with ’str’, as shown in Figure 4. By converting different types of input into different word tags, the number of tags is reduced and the difficulty of feature acquisition is reduced.

5.Feature enhancement. In order to further reduce the training difficulty of the model, accelerate the training speed and improve the ability of feature extraction, in the preprocessing, the combination of markers that usually appear together can be replaced with fixed word markers according to the safety experience. For example,the conditional statement commonly seen in SQL injection attacks shown in Figure 5 is replaced with ’equation’.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Articol SQLi using artificial networks

















