Research progress of Web vulnerability mining based on machine learning

Zhengshuang Shi
The University of Edinburgh School of informatics Edinburgh, UK
shizhengshuang@ed.ac.uk

Abstract. Web is an area of continuous development and innovation every year. It not only changes the way people deal with personal, social contact and business, but also ushers in new opportunities in the era of Internet. Network security develops rapidly. Web security also needs new methods to solve. Machine learning technology is used to detect attacks, and its algorithm model is used to predict attacks. This kind of method that can automatically update the module to adapt to the new attack can greatly reduce the manpower consumption. This paper sorts out the methods of mining typical web attack vulnerabilities using machine learning. The future development trend is given.

Keywords: Machine learning, Web attack, Vulnerability detection, Network security.

1. Overview
With the continuous development of information technology, more and more industries, such as banking, transportation, manufacturing, commerce and education, are moving to the network and providing services in the form of Web[1]. While network attacks are becoming more and more diversified, more security vulnerabilities are disclosed. The vulnerability mainly affects the confidentiality, integrity and availability of web applications. Web vulnerability usually refers to the vulnerability of the website, which may be caused by the poor consideration of the coder when writing the code. The common web vulnerabilities include SQL injection, cross site scripting (XSS) and so on. If there is a vulnerability in the website and it is exploited by an illegal attacker, the attacker can easily control the whole website, and further enhance the right to obtain the permission of the website server and then control the whole server. Because of the site widely accessible, the security level of the site must always be maintained. Literature detected a certain range of Web sites, and found that as many as 60% of the sites have a maturity level lower than 3 [2]. This shows that there are many vulnerabilities in the web site. Therefore, it is of great significance to detect web application security vulnerabilities.

Vulnerability mining can be used as an active defense measure, and can reduce the loss caused by the possible threat. The security vulnerability of the specified remote or local computer system is detected by scanning and other means, and exploitable vulnerabilities are found. There are three methods for vulnerability mining, namely, static mining based on source code, dynamic mining based on target code and dynamic-static mining [3]. The security of web program can be effectively improved by timely patching the vulnerabilities. Traditional detection methods are not suitable for the current large web applications, especially for the low detection rate of hidden security vulnerabilities and high false positives, which affect the overall security of the network [4]. It is a main method to improve the...
detection efficiency to apply machine learning (ML) method to web vulnerability mining. By executing ML model and learning algorithm, the data set with characteristics is established, and the mining process is optimized to reduce the loss of system mining [5].

2. XSS attack mining technology
XSS is one of the most common application layer attack technology. It allows attackers to embed malicious JavaScript, VBScript, ActiveX, HTML, etc. into vulnerable dynamic pages, cheats users, and executes scripts on their computers to collect data [6]. The essence of XSS attack is that web application does not check the content for malicious code before running in the user's browser. For example, XSS attack for the storage reads cookies:

```
var img = document.createElement('img')
img.src = 'http://www.xss.com?cookie=' + document.cookie
img.style.display = 'none'
document.getElementsByTagName('body')[0].appendChild(img)
```

Although several methods have been introduced to mitigate XSS attacks so far, it is still a real problem. With the increase of attack cases, intruders are introducing more complex methods to embed scripts to cheat users. Based on machine learning method, static and dynamic features of malicious and benign samples are extracted, and classification tools are established to realize web vulnerability mining [7].

Literature uses unsupervised learning method to detect cross site scripting vulnerabilities. This method focuses on balancing the load between the client and the server. First, the client carries out decentralized initial detection [8]. Once the suspicious degree exceeds a certain threshold, it will be discarded. Otherwise, it will be forwarded to the server agent for further processing. At the same time, the agent uses attribute clustering and ranking aggregation technology to inspect JavaScript code in detail. Wu et al. summarized six features by manual selection and mathematical statistics, trained and classified with support vector machine (SVM) algorithm, and realized the detection of cross site script vulnerabilities [9]. The implementation cases of support vector machine algorithm are as follows:

```python
# encoding=utf-8
import time
import numpy as np
import pandas as pd
from sklearn.cross_validation import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import datasets
from sklearn import svm
if __name__ == '__main__':
    print('prepare datasets...')
    # Iris data set
    # iris=datasets.load_iris()
    # features=iris.data
    # labels=iris.target
    # MNIST data set
    raw_data = pd.read_csv('../data/train_binary.csv', header=0)
    data = raw_data.values
    features = data[:, 1::]
    labels = data[:, 0]
    train_features, test_features, train_labels, test_labels = train_test_split(features, labels, test_size=0.33, random_state=0)
    time_2=time.time()
    print('Start training...')
```
clf = svm.SVC()  # svm class
clf.fit(train_features, train_labels)  # training the svc model
time_3 = time.time()
print('training cost %f seconds' % (time_3 - time_2))
print('Start predicting...')
test_predict = clf.predict(test_features)
time_4 = time.time()
print('predicting cost %f seconds' % (time_4 - time_3))
score = accuracy_score(test_labels, test_predict)
print("The accuracy score is %f" % score)

Detection method based on machine learning, through collecting a large number of data, uses ML algorithm to extract the knowledge about the vulnerability, so as to detect the existence of the vulnerability. The characteristic of machine learning method is that the model training cost is large, and it is very fast and efficient in the application model detection stage, which is a big advantage compared with the test based detection method. However, it also has the problem of false positives.

3. SQL injection mining technology

SQL language, namely structured query language, provides a method to operate database data. The website background usually uses user input information to dynamically construct SQL statements to interact with the back-end database. When the website background does not filter user input reasonably and uses it directly, it is easy to produce SQL injection vulnerability [10]. For example, the SQL command with ?Id=1 and 1=1 is:

```
Select * from article where id=1 or 1=1,
```

This statement queries the records of the entire table. Traditional SQL injection vulnerability mining technology mainly includes static analysis, dynamic analysis and hybrid analysis of dynamic and static. But the traditional technology is time-consuming in building the mining environment, and has a high rate of missing reports, and is subject to the constantly updated system software and the loopholes that have been exposed in succession, so it is difficult to have further room for improvement. By using historical vulnerability data modeling through machine learning, we can understand code semantics more deeply, and improve the accuracy and recall rate of SQL injection vulnerability mining.

In order to solve the impact of imbalance problems on ML algorithm, Hu used undersampling method to randomly select two times the number of normal code samples, and used support vector machine algorithm combined with bag of words model to realize the automation of SQL injection vulnerability mining[11]. Jiang combines real AdaBoost algorithm and FP growth algorithm, and proposes a SQL injection detection technology that works in HTTP application layer and can detect potential threats only by analyzing user input without knowing anything about background program[12]. But in the preprocessing part, the recursive decoding algorithm is inefficient, time-consuming, needs to simplify the algorithm, and has insufficient adaptability to different databases. The FP growth algorithm is as follows:

**Construct FP-Tree:**

(a) Scan transaction database D once. Collect the set F of frequent items and their support. F is sorted in descending order of support, and the result is frequent item table L.

(b) Create the root node of FP-Tree and mark it with "null". For each transaction trans in D, execute: select the frequent items in trans and sort them in the order in L. Let the sorted frequent item table be [p | P], where p is the first element and P is the table of the remaining elements. Call insert_tree([p | P], T).

The implementation of this process is as follows:

If T has children N such that n.item-name = p.item-name, the count of N will be increased by 1; otherwise, create a new node N, set its count to 1, link it to its parent node T, and link it to the node with the same item name through the node chain structure. If P is not null, insert is called recursively_tree(P, N).

FP-Tree mining is implemented by calling FP_growth(FP_tree, null). The process is as follows:
procedure FP_growth(Tree, α)
if Tree contains a single path P then
for Every combination of nodes in path P (denoted as β)
The pattern β ∪ α is generated, and β is the minimum support degree of nodes;
else for each ai In the head of Tree {
A pattern β = ai ∪ α is generated, and its support is support = ai.support;
The conditional pattern base of β is constructed, and then the conditional FP-Tree of β is constructed;
if Treeβ ≠ ∅ then
    call FP_growth (Treeβ, β);
}

4. Webshell attack mining technology
Webshell is a command execution environment in the form of ASP, PHP, JSP or CGI, which can also be called a web backdoor. After invading a website, lawbreakers usually mix the ASP or PHP backdoor file with the normal web page file in the web directory of the website server. Then they can use the browser to access the ASP or PHP backdoor and get a command execution environment to control the website server. Webshell is also often referred to as the right of an intruder to operate on a web server through a web port to some extent. For example, webshell file upload attack can bypass mime detection by modifying the content type of HTTP header.

```plaintext
POST /upload.do HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Host: localhost
Content-Type: multipart/form-data; boundary=xYzZY
Content-Length:155
Content-Disposition: form-data; name="userfile"; filename="shell.jsp"
Content-Type: image/gif (Content-Type: text/plain)
<% system($request.getParm['command']);%>
```

With the development of attack technology, many malicious webshell are difficult to be detected by the conventional detection method. Attackers use various ways of confusion and mutation to make malicious webshell files disguise as normal web pages. Because a large number of long strings with unknown meaning will be generated, the readability of the code will be greatly reduced, so the traditional detection methods are easy to fail. Zhang used to convert the source code of the page into the operation code at the bottom of the machine, and then read the operation code string into the text, using the text processing method in natural language processing, through word2vec The algorithm learns from the text data of training samples to obtain the high-dimensional word vector corresponding to each opcode, and then converts each text into the corresponding number vector matrix, which is used as the input data set of ML and deep learning model[13].

Hu et al.[14] proposed a supervised webshell detection model based on decision tree collective learning algorithm. They extract features from three levels: the overall attribute of the document, the basic attribute of the code level and the function attribute of the dynamic call page, and then use the decision tree algorithm to learn the features, and use the collective learning method to improve, so as to avoid the problem of over fitting. The decision tree pseudo codes are as follows:

```
Training sample set D={(x1,y1),(x2,y2)……(xn,yn)}
Attribute set A={a1,a2,……, an}
TreeGenerate(D,A):
    Generate node
    if All samples in D belong to the same category C:
        Mark the node as category C leaf node
        return
    end if
    if Attribute set A is empty or all attribute values of D are the same:
```

Mark node as the largest class
return
end if
Select the best partition attribute A from a*
for a in a*:
Generate a branch for node, and let DV represent the sample subset with a* attribute value of a in D
if DV is empty:
    continue;
else:
    TreeGenerate(Dv,A\{a*\})Recursion continues
end if
end for

Aditya Kuppa et al. proposed a patent for webshell detection, which analyzed the source code, extracted the static features including information entropy, coincidence index, the number of system function calls, compression ratio, the longest character length, and file signature, and carried out dynamic testing in the sandbox, extracted the dynamic features such as the number of system function calls and network traffic, and finally used ML to train the classifier[15].

5. Some directions in the future

5.1. Data set construction
Data set is the soul of machine learning. The quantity of data in training data set is directly proportional to the performance of trained model. The quality and quantity of data set seriously affect the performance of trained model. In fact, however, it is difficult to obtain high quality and large number of web vulnerability data sets. In order to obtain a high-quality training data set, we need to collect vulnerability information of various open source web applications, and then we need to analyze, locate, track and mark these vulnerability information, which is a huge project. Therefore, how to solve the problem of training data set is the first problem.

5.2. Cross project
There are three problems that hinder the application of machine learning in web vulnerability mining. For the new web applications, because there is no historical version, there is no historical vulnerability, there are not enough vulnerability codes as training samples to support machine learning training, and the vulnerability mining model cannot be built for the specified program. The number of vulnerabilities in the historical version of the application is small, and it is still not enough to train a high-performance vulnerability mining model. Due to the different web applications, different developers' development habits and frameworks lead to different vulnerability distribution. Therefore, the vulnerability code of one application cannot be used as a training sample for machine learning of another application. Migration learning algorithm [14] can solve the above problems to a certain extent. For the new web applications without historical vulnerabilities, we can use the existing vulnerability prediction model for a certain type of web applications to fine tune their vulnerability mining. For the web applications with a small number of vulnerabilities, we can find several similar projects to synthesize their vulnerabilities data set training, and then through the model to predict the corresponding web applications.

5.3. Mining visualization
In order to show the mining process and the security of web applications better, more intuitive understanding of the performance of the algorithm and the degree of abnormal data processing, we need to combine visualization technology. Real time analysis of abnormal data sources, and show the vulnerabilities, through the analysis of vulnerability threats, compared with the threat of artificial analysis, more accurate grasp of the potential danger of vulnerability. With the increase of anomaly
types, it increases the difficulty of real-time detection, records the running time of the algorithm, and visualizes the processing. It is used to find the difficulty of the algorithm model to deal with these data, so as to improve the use of the algorithm.

6. Conclusions
With the increasing attacks on Web applications, the network security situation is becoming more and more critical. In this paper, the research of using machine learning to deal with multiple web vulnerabilities is sorted out. Through the comparison of different methods, this paper introduces the traditional web vulnerability mining technology and its shortcomings, and highlights the advantages of using machine learning to mine web vulnerabilities in the big data environment. Web vulnerability mining based on machine learning needs to be further developed and improved. The hot topics are data set construction, cross project research and visual display.

References
[1] Amankwah R, Chen J, Kudjo P K, et al. An empirical comparison of commercial and open: source web vulnerability scanners. Software Practice and Experience, 2020.
[2] Mantra I, Hartawan M S, Saragih H, et al. Web Vulnerability Assessmnt and Maturity Model Analysis on Indonesia Higher Education. Procedia Computer ence, 2019, 161, pp.1165-1172.
[3] Pu Z Y. Analysis of Computer Security Vulnerability Detection and Vulnerability Repair [J]. China Computer & Communication, 2020, 32, pp.206-208.
[4] Sun X Z, Fan C L, Lu Y, Xu K, Leng X J, Luan W P. Automatic detection method of network security vulnerabilities based on artificial intelligence. Automation & Instrumentation, 2020, 06 pp.180-183.
[5] Song Y N, Liu P. Research on vulnerability detection technology based on machine learning optimization strategy. Information Technology, 2018(02), pp. 37-42+47.
[6] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, “Botnet in ddos attacks: trends and challenges,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2242–2270, 2015.
[7] D. K. Bhattacharyya and J. K. Kalita, Network anomaly detection: A machine learning perspective, CRC Press, 2013.
[8] Goswami S, Hoque N, Bhattacharyya D K, et al. An unsupervised method for detection of XSS attack. International Journal of Network Security, 2017, 19.
[9] Wu S H, Cheng S B, Hu Y. Web Attack Detection Method Based on Support Vector Machines. Computer Science. 2015, 42(s1), pp. 362–364.
[10] SHAR L K, TAN H B K, BRIAND L C. Mining SQL Injection and Cross Site Scripting Vulnerabilities Using Hybrid Program Analysis[C]/IEEE. ICSE'13 Proceedings of the 2013 International Conference on Software Engineering, May 18 - 26, 2013, San Francisco, CA, USA. New York: IEEE, 2013.pp. 642-651.
[11] Hu J W, Zhao W, Yan Z, Zhang B. Analysis and Implementation of SQL Injection Vulnerability Mining Technology Based on Machine Learning. Netinfo Security, 2019(11), pp. 36-42.
[12] Jiang L. Research on SQL detection technology based on machine learning. Nanjing University of Posts and Telecommunications, 2017.
[13] Zhang H. Research on Webshell Detection Method Based on Machine Learning. Shanghai Jiao Tong University, 2019.
[14] Hu J K, Xu Z, Ma D H, Yang J. Research of Webshell Detection Based on Decision Tree. Journal of network new media. 2012, 1(6), pp. 15-19.
[15] Kuppa A,Keyes O, Kennedy J, et al. Web shell detection . Google Patents. 2018.
[16] Ghaffarian S M, Shahriari H R. Software Vulnerability Analysis and Discovery Using Machine-learning and Data-mining Techniques: A survey. ACM Computing Surveys(CSUR), 2017, 50(4), pp. 56.