"Experimental setup for Detecting SQLI Attacks using Machine Learning Algorithms"

Vinitha Hannah Subburaj, Ph.D., Assistant Professor
Research student: Binh An Pham
School of Engineering, Computer Science and Math, West Texas A&M University, Canyon, Texas – 79016
E-mail: vsubburaj@wtamu.edu, Office Phone: +1. 806. 651. 5264

Abstract

This research effort aimed at investigating and looking at different ways to protect websites from SQL injection attacks. In this research effort, machine learning algorithms were used to detect such SQLi attacks. Machine Learning (ML) algorithms are algorithms that can learn from the data provided and infer interesting results from the dataset. We have used SQL code as our data and ML algorithms to detect malicious code. The machine learning model developed in this research effort can detect such attacks from happening in future. The precision and accuracy of the machine learning algorithms in terms of predicting the SQLi attacks has been calculated and reported in this research paper.

Theoretical Background

Cybersecurity is a prevailing issue across the nation. In the twenty first century, almost everyone around the globe is using at least one of the internet websites that contain his/her private information. Since privacy concerns us, this research effort focuses one of the most recent cyber-attacks which is the SQL injection (SQLi) attacks. As a result of SQL injection attack on websites, data could be destroyed, stolen, or manipulated. SQL injection attacks are done by injecting despicable SQL statements through the entry field of the website or the application; thus, manipulating the database. SQL injection attacks had proven their danger on several website types such as social media, e-shopping, etc... In order to prevent such attacks from occurring, this research effort investigates on efficient ways of detection and prevention, so we can preserve each cyber-user's right of privacy.

Methodology

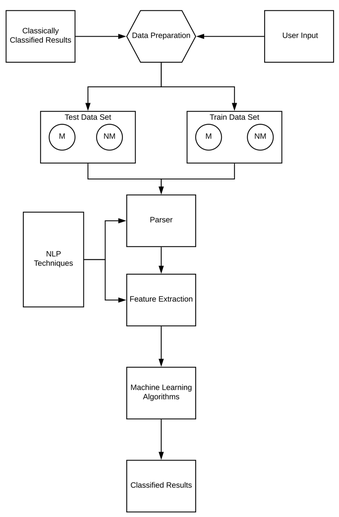

Figure 1. Experimental Model Diagram

Technology used:

- Python for data processing
- Scikitlearn, pandas, and xgboost for model setup

Data Processing

```
470                -@@ union select 1,version() -- 1
363                  union select \.#,@@VERSION -- 1'
564                              https://www.taobao.com
606                               https://www.mail.ru
445           or 1#"@ union select 'a',version() -- 1
514   Mind numbing: This game makes you do the same ...
4               or (\+)=1 union select 1,@@VERSION -- 1
3     or (1)=1 union select 1,banner from v$version ...
726                                              HUNGO
64                                     or 1=1 -- 1
```

Figure 2. Snapshot of training dataset

- Figure 2 is the snapshot of processed and randomized training data for the model.

```
810                                        6478371333
112                  union select +!<@,version() -- 1
127              +$ . union select null,@@VERSION -- 1
812                                        6478567790
293             or ""<@ union select 1,version() -- 1'
546   Barbie as Rapunzel: A Creative Adventure: I pu...
453                              or "(1"=1 or 1=1 -- 1
820                                        6462146987
823                                        6474876638
177                                  &!<@ or 1=1 -- 1
```

Figure 3. Snapshot of testing dataset

- Figure 3 is the snapshot of processed and randomized testing data for the model.

Results

Classification report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	76
1	1.00	1.00	1.00	99
accuracy			1.00	175
macro avg	1.00	1.00	1.00	175
weighted avg	1.00	1.00	1.00	175

Confusion Matrix

	Predicted Class 0	Predicted Class 1
Class 0	76	0
Class 1	0	99

Figure 4. Classification report and Confusion matrix for Logistic Regression algorithm

- Figure 4 is a report for one of the algorithms used in this experimental model.
- Classification portion shows the accuracy and the number of support datapoints.
- Confusion matrix shows the classification performance, false and true positives, and false and true negatives.

Conclusion

1) An experimental setup to run different machine learning algorithms to detect SQL Injection attacks was developed.
2) Research results produced can be used by the research community working on Cyberattacks.
3) Accuracy of the machine learning algorithms used in research were determined.
4) Research has the potential to be expanded in future by adding more machine learning algorithms.

References

[1] Acunetix. "What Is SQL Injection (SQLi) and How to Fix It." Web.
[2] Tajpour, Atefeh, Suhaimi Ibrahim, and Mohammad Sharifi. "Web application security by sql injection detectiontools." IJCSI International Journal of Computer Science Issues 9.2 (2012): 332-339.
[3] Vinitha Subburaj, Daniel Thomas Loughran, Mayar Kefah Salih, "All About SQL Injection Attacks", CISSE 2018, New Orleans, LA.
[4] McClure, Russell A., and Ingolf H. Kruger. "SQL DOM: compile time checking of dynamic SQL statements." Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on. IEEE, 2005.
[5] Boyd, Stephen W., and Angelos D. Keromytis. "SQLrand: Preventing SQL injection attacks." International Conference on Applied Cryptography and Network Security. Springer, Berlin, Heidelberg, 2004.
[6] Halfond, William GJ, and Alessandro Orso. "AMNESIA: analysis and monitoring for NEutralizing SQL-injection attacks." Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. ACM, 2005.
[7] Alwan, Zainab S., and Manal F. Younis. "Detection and Prevention of SQL Injection Attack: A Survey." (2017).
[8] Joshi, A., & Geetha, V. (2014). SQL Injection detection using machine learning. Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on, 1111-1115.
[9] Valeur, Fredrik, Darren Mutz, and Giovanni Vigna. "A learning-based approach to the detection of SQL attacks." International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Berlin, Heidelberg, 2005.
[10] Ladole, Aniruddh, and DA, Phalke. "SQL Injection Attack and User Behavior Detection by Using Query Tree Fisher Score and SVM Classification." International Research Journal of Engineering and Technology 3.6 (2016).
[11] Rawat, R. & Kumar, S. (2012). SQL injection attack detection using SVM. International Journal of Computer Applications.
[12] https://github.com/client9/libinjection.git
[13] https://archive.ics.uci.edu/ml/machine-learning-databases/00237
[14] https://www.kaggle.com/wjburns/common-password-list-rockyoutxt
[15] https://www.kaggle.com/hackerrank/developer-survey-2018
[16] https://www.kaggle.com/siddharthkumar25/malicious-and-benign-urls
[17] https://scikitlearn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support
[18] https://scikitlearn.org/stable/modules/generated/sklearn.metrics.confusion_matrix

Contact Information

Vinitha Hannah Subburaj
Assistant Professor of Computer Science
West Texas A&M University
vsubburaj@wtamu.edu

Binh An Pham
Research Assistant
West Texas A&M University
bpham1@buffs.wtamu.edu

Acknowledgements

Funding was provided by West Texas A&M University
and the Killgore Faculty Research program.