International Journal of Research in Engineering, Science and Management
Volume 6, Issue 4, April 2023
https://www.ijresm.com | ISSN (Online): 2581-5792

Auto Encoder System for Intrusion Detection

Kalagara Tharakaram[1*], Kursange Tharun Kumar[2], Saba Sultana[3]
[1,2]B.Tech. Student, Department of Computer Science and Engineering, CMR Technical Campus, Hyderabad, India
[3]Assistant Professor, Department of Computer Science and Engineering, CMR Technical, Hyderabad, India

***Abstract*: Monitoring a web application for attacks and issuing alerts when one is detected is the job of an intrusion detection system. In contrast, existing implementations are time-consuming and require a thorough understanding of security domains. A web application is an easy target for cyber-attacks due to its vulnerability and network accessibility. To begin with, we examine the feasibility of an unsupervised/semi-supervised method for detecting web attacks based on the Robust Software Modelling Tool (RSMT), which monitors and characterizes web applications in runtime automatically. In the second step, we describe how the RSMT encodes and reconstructs the call graph using a stacked denoising auto encoder. Finally, both datasets were tested using the RSMT and the results were analyzed. Little labelled data can be used to detect attacks is efficient and accurate when we use the Long Short Term Memory algorithm.**

***Keywords*: Intrusion detection system, semi supervised, RSMT, autoencoder.**

1. Introduction

Web attacks refer to the malicious activities that target web applications, servers, and clients to gain unauthorized access, disrupt normal operations, or steal sensitive information. Traditional security measures such as firewalls, intrusion detection/prevention systems, and antivirus software are often inadequate in detecting and preventing these attacks, as attackers continually evolve their methods and use sophisticated techniques to evade detection. In recent years, deep learning has emerged as a promising approach to detecting web attacks. Deep learning is a subset of machine learning that involves the use of neural networks with multiple layers to learn automatically and extract features from large datasets. With its ability to learn complex patterns and detect anomalies, deep learning can help identify and mitigate a wide range of web-based threats.

The key advantage of deep learning is its ability to learn and adapt to new attack patterns, making it an effective tool for detecting zero-day attacks. It can also analyze a large volume of data in real-time, providing near-instantaneous detection and response to attacks. To use deep learning for web attack detection, researchers typically build models based on large datasets of web traffic and use various techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to identify patterns and anomalies in the data. These models can then be integrated into existing security systems to enhance their effectiveness. Detecting web attacks using deep learning is a promising and rapidly evolving field, with the potential to improve the security of web applications significantly and services. It is critical to defend network systems and information assets from network attacks, and there exist various techniques to deal with network attacks. Among those, public key cryptography and digital certificates can be used to protect network systems by enabling source authentication. These cryptographic techniques make it possible to verify whether network traffic is originated from a trusted source or not. Thus, we can filter out malicious traffic from untrusted sources.

2. Literature Survey

Detecting Web Attacks with End-to-End Deep Learning. Authors: Yao Pan, Fangzhou Sun, Jules White, Douglas C. Schmidt, Jacob Staples, and Lee Krause. The paper outlines an unsupervised end-to-end deep learning approach that was implemented to automatically identify attacks on web applications. The architecture and results of this approach are discussed. To assess the efficacy of this intrusion detection system, the authors developed a number of test applications and synthetic trace datasets and measured the performance of unsupervised learning in detecting attacks on these datasets.

A classification of SQL-injection attacks and countermeasures. Authors: Halfond WG, Viegas J, Orso A.

SQL injection attacks can have severe consequences for web applications, as they can provide attackers with unrestricted access to the databases that support these applications, and to the potentially sensitive information stored within them. Despite numerous attempts by researchers and practitioners to address the issue of SQL injection, current approaches either do not fully address the problem or are limited in their effectiveness, hindering their widespread adoption.

An adaptive network intrusion detection method based on PCA and support vector machines. Advanced Data Mining and Applications. Authors: Xu X, Wang X.

Computer security heavily relies on network intrusion detection, but the current intrusion detection systems (IDSs) often fall short due to the constant development of new attacks and the fast-paced increase in network traffic volumes. To overcome these limitations and improve the accuracy and speed of IDSs, this paper presents a new adaptive intrusion detection method that leverages principal component analysis (PCA) and support vector machines (SVMs).

*Corresponding author: venkatprojects768@gmail.com

Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. Author: Pietraszek T.

In order to monitor computer systems for potential security breaches, Intrusion Detection Systems (IDSs) are utilized. When signs of security violations are detected by IDSs, alerts are generated to notify human analysts, who then evaluate the alerts and determine the appropriate course of action. However, in practice, IDSs often generate an overwhelming number of alerts, with the majority of these alerts being false positives - alerts that are triggered by benign events and not indicative of a security breach.

An anomaly detection method to detect web attacks using stacked auto-encoder. Authors: Ali Moradi Vartouni, Saeed Sedighian Kashi, Mohammad Teshnehlab

Information security is currently facing significant threats from network-borne attacks. To counter these attacks, a range of measures have been implemented, including scanners, encryption devices, intrusion detection systems, and firewalls. Web application firewalls, which use intrusion detection techniques to safeguard servers against HTTP traffic, have also been developed. Additionally, machine learning algorithms have been employed to detect anomalies in these firewalls and enhance their effectiveness.

3. Proposed System

Detect attacks from web application using Deep Learning Network and Robust Software Modelling Tool (RSMT). RSMT tool is a web monitoring tool which monitor execution behaviour of web application and record in a trace file. Trace file contains low dimensional raw data and it cannot be used for Deep Learning Network. To convert this raw data to deep learning features we are using autoencoder technique. Auto encoder will convert raw data into deep learning features. These features will be passes to propose Autoencoder algorithm which will generate train and test data from features. Autoencoder algorithm requires un-label train data to generate the model and new test data will be applied to the Autoencoder train model to identify new test data is a normal request or contains an attack. If new test data not available in the Autoencoder train model then it will be considered an attack.

4. System Architecture

A system architecture typically includes many different layers or tiers, each with its own specific functions and responsibilities. For example, in a web-based system, the architecture might include a presentation layer (the user interface), a business logic layer (which handles data processing and storage), and a data access layer (which interacts with the database).

The architecture of a system is typically documented using a variety of diagrams and models, such as block diagrams, flowcharts, and UML (Unified Modelling Language) diagrams. These diagrams help to illustrate the relationships and dependencies between the different components of the system, and provide a clear understanding of how the system works.

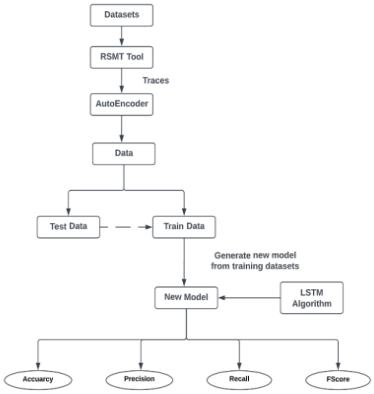

Fig. 1. System architecture

5. Implementation

A. SVM Algorithm

SVM can also be used for detecting web attacks. the SVM model is trained on a set of labelled data, where each instance represents a web request or transaction and is labelled as either a normal or malicious request. The input features for each instance may include things like the requested URL, HTTP method, user agent, IP address, and other relevant attributes.

The SVM model then learns to classify new incoming requests as either normal or malicious based on the learned classification boundary. This can help in identifying and blocking potentially harmful requests before they can cause damage to the web application or system.

B. Naive Bayes Algorithm

The Naive Bayes algorithm works by first estimating the prior probabilities of the two classes (normal and malicious) based on the training data. It then estimates the conditional probability of each feature given the class label using the training data. These probabilities are combined using Bayes' theorem to calculate the posterior probability of each class given the feature values.

During classification, the algorithm calculates the posterior probability of each class given the feature.

C. Auto Encoder Algorithm

An autoencoder can be trained on a large set of normal network traffic data to learn the underlying patterns and structures in the data.

The algorithm works by first encoding the input data into a compressed representation using a neural network. The encoder network reduces the dimensionality of the input data by transforming it into a lower-dimensional space. The compressed representation is then decoded back into the original input data using a decoder network. The decoder network tries to reconstruct the original input data from the compressed representation.

During training, the autoencoder is trained to minimize the difference between the original input data and the reconstructed data. The objective is to learn a compressed representation that captures the most important features of the input data while minimizing the reconstruction error.

During testing, the autoencoder is used to detect anomalies in the input data. Anomalies are detected by measuring the reconstruction error between the original input data and the reconstructed data. If the reconstruction error exceeds a certain threshold, the input data is classified as anomalous and may indicate the presence of web attacks.

D. Long Short-Term Memory Algorithm

The LSTM architecture consists of multiple memory cells that can store information over a long period of time. The memory cells are connected through gates that control the flow of information into and out of the cells. The gates are controlled by activation functions that determine the amount of information that is retained or discarded.

During training, the LSTM model is trained to minimize the difference between the predicted output and the actual output. The objective is to learn a model that can accurately predict the next output in the sequence given the previous inputs.

During testing, the LSTM model can be used to detect anomalies in the network traffic data. Anomalies are detected by comparing the predicted output with the actual output and measuring the difference between them. If the difference exceeds a certain threshold, the input data is classified as anomalous and may indicate the presence of web attacks.

6. Conclusion

LSTM algorithms have been found to be effective in detecting web attacks based on the results of this study. In addition to its high performance, the LSTM algorithm has the advantage of being able to process sequential data, which is particularly relevant for detecting web attacks. The LSTM algorithm is also capable of learning from previous inputs, making it well-suited for handling dynamic and evolving attacks.

Acknowledgement

We thank CMR Technical Campus for supporting this paper titled “Auto encoder system for intrusion detection”, which provided good facilities and support to accomplish our work. Sincerely thank our Chairman, Director, Deans, Head of the Department, Department of Computer Science and Engineering, Guide and Teaching and Non- Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work**.**

References

[1] Al-Fayoumi, M. A., Al-Naffouri, T. Y., & Al-Salman, A. S. (2019). Deep learning for intrusion detection: A comprehensive review. IEEE Communications Surveys & Tutorials, 22(3), 1380-1419.

[2] Alom, M. Z., Yakopcic, C., Hasan, M., Taha, T. M., & Asari, V. K. (2019). Intrusion detection using deep learning: A review. IEEE Access, 7, 45400-45419.

[3] Kim, T. H., Park, J. H., Lee, J. H., & Choi, D. H. (2016). A deep learning approach to network intrusion detection. Journal of Information Security and Applications, 31, 1-12.

[4] Mirsky, Y., Shabtai, A., & Elovici, Y. (2018). Network-based detection of web application attacks using deep autoencoder neural networks. IEEE Transactions on Information Forensics and Security, 13(10), 2560-2573.

[5] Puniya, P., & Lamba, G. (2019). Deep learning-based approaches for intrusion detection system: A review. Journal of Intelligent & Fuzzy Systems, 37(4), 4997-5014.

[6] Al-Harthi, A. S., Shamsuddin, S. M., & Sulaiman, M. N. (2018). A deep learning approach for detecting SQL injection attacks in web applications. Journal of Ambient Intelligence and Humanized Computing, 9(4), 1121-1134.

[7] Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., & Asari, V. K. (2019). Intrusion detection using deep belief networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).

[8] Gao, S., Liu, X., Wu, C., & Chen, X. (2020). A hybrid deep learning approach for web attack detection. IEEE Transactions on Industrial Informatics, 17(2), 1107-1116.

[9] Ghiasi, S., El-Khatib, K., & Gandomi, A. H. (2021). A comprehensive review of deep learning techniques in cyber-security applications. IEEE Access, 9, 13806-13831.

[10] Thanh, T. L., Trung, D. T., & Phuong, T. M. (2021). Web attacks detection based on deep learning with data augmentation. Journal of Ambient Intelligence and Humanized Computing, 12(4), 3907-3920.