IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115171

DEEP LEARNING SYSTEM TO INTRUSION DETECTION BASED ON RECURRENT NEURAL NETWORK

Narmada B[1], Brinda S[2], Prasanna S[3], Shneka P[4]

[1]Assistant Professor and Head of the Department, Department of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, Salem, Tamil Nadu, India

[2,3,4]Student, Department of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, Salem, Tamil Nadu, India

Abstract: Recently, the huge amounts of data and its incremental increase have changed the importance of information security and data analysis systems. Intrusion detection system (IDS) is a system that monitors and analyzes data to detect any intrusion in the system or network. High volume, variety and high speed of data generated in the network have made the data analysis process to detect attacks by traditional techniques very difficult. To proposed Recurrent Neural Network (RNN) algorithm to detect the IDS. The data processed by the preprocessing module are compressed by the auto-encoder module to obtain a lower-dimensional reconstruction feature, and the classification result is obtained through the classification module. Compressed features of each traffic are stored in the database module which can both provide retraining and testing for the classification module and restore these features to the original traffic for post event analysis and forensics. We used KDD cup 99 to train and test the model. Through this way, we could reduce the number of false alarms and increase the accuracy of the designed intrusion detection system.

Keywords: Intrusion detection system, Recurrent Neural Network, KDD cup99

I. INTRODUCTION

One of the main security threats of intruder is a computer and network environment. Intruder (or) malware is designed to deliberately infiltrate or damage a computer system without the owner's knowledge. It can appear in the form of code, scripts, active content and other software. That is, computers that are compromised with malware are often networked together to form botnets, and many attacks are launched using these malicious, attacker-controlled networks. Botnets are then used by miscreants to launch denial of service attacks, send spam emails, or host scam pages.The amount of malware threats on the Internet has increased significantly over the past few years. Hence the traditional methods of malware detection do not sufficient. The proposed algorithm is very efficient in compression of the previous method. Recurrent Neural Network (RNN) algorithms for analyzing network user behavior. RNN algorithm generates detection model learning from a sufficient dataset of malicious software. Signature-based detection techniques detect unknown malware.

II. SYSTEM STUDY

A. Exisiting System

An existing novel approach for intrusion detection that uses Support Vector Machine (SVM) and decision tree. Existing to exploit quantitative data flow properties to extract highly characteristic behavior patterns from collections of known intrusion. By combining a SVM and NB based machine learning techniques provide low classification accuracy prediction results.

Disadvantages:

- ✓ The computational complexity of this current method would be limiting in a real-time setting.
- ✓ Increasing the complexity of detection makes for a much more robust analysis system.
- ✓ In the graph mining data representation gets away from the need to specify the appropriate.

B. Proposed System:

It focuses on dynamic analysis to discover hidden behavior in packed samples where it is essential to do so. The proposed RNN algorithm gives high accuracy for intrusion detection. In addition, it also classifies malware based on their

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022
DOI: 10.17148/IJARCCE.2022.115171

family and checked the accuracy of each of the malware behavior. Static analysis targets source and object codes and examines codes without actually starting a project. It decompiles malware source code to detect commands, reports, and vulnerabilities in many programs. Dynamic analysis is a method of searching for certain types of memory leakage, traffic flow, and flows data into code that actually runs. However, a large amount of storage is required for applying this method to the mobile environment, and the performance overhead is high for system matching.

Advantages:

✓ Very effective at detecting known threats.

✓ The detection of attacks by looking for specific patterns, such as byte sequences in network traffic, or known malicious instruction sequences used by malware.

✓ Unique identifier is established about a known threat so that the threat can be identified in the data.

III.SYSTEM REQUIREMENTS

A. Hardware Requirements:

CPU type : Intel core i3
Ram size : 8 GB
Hard disk capacity : 500 GB

B. Software Requirements:

Operating System : Windows 10
Language : Python
Tool : Anaconda

IV.MODULE DESCRIPTION

A. KDDcup99 dataset

KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one
specific attack type.

KDD dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. The Dataset contains various attacks such as Denial of Service Attack (DOS), User to Root Attack (U2R), Remote to Local Attack (R2L) and Probing Attack.

✓ Denial of Service (dos): Attacker tries to prevent legitimate users from using a service.

✓ Remote to Local (r2l): Attacker does not have an account on the victim machine, hence tries to gain access.

✓ User to Root (u2r): Attacker has local access to the victim machine and tries to gain super user privileges.

✓ Probe: Attacker tries to gain information about the target host.

B. Pre-processing

Pre-processing of data becomes crucial in avoid the noisy, missing and inconsistent data available in the dataset. Since the records of the dataset are collected from multiple and heterogeneous sources, the quality of the data deteriorates and therefore needs to be pre-processed. Several factors affect the quality of the data. These factors comprise accuracy, completeness, consistency, timeliness, believability and interpretability. The proposed pre-processing check null values, and fill missing values in efficient manner.

C. Feature selection

Feature selection aims to select the best feature in the data set. Deep learning technique can classify the data into a set of class features and targets. Feature selection (variable elimination) helps understand the data, reduces computing needs, reduces dimensional curse effects and improves the performance.

D. RNN

The RNN architecture is the addition of sequential information to the feed forward neural network. The RNN performs the same task for each part. This is why it is called a recurrent network; the output is dependent upon the previous computation.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115171

V. CONCLUSION

The advent of network based technologies has increased the associated vulnerabilities. As a result, it has become paramount to design and implement effective IDS. In this paper, we apply feature selection methods to improve our understanding of relevant features inside network traffic data and construct potent detection systems using KDDcup99 dataset. The proposed Recurrent Neural Network (RNN) algorithm to detect the IDS. The data processed by the preprocessing module are compressed by the auto-encoder module to obtain a lower-dimensional reconstruction feature, and the classification result is obtained through the classification module.

REFERENCES

[1] Ashwini Mujumdar, Gayatri Masiwal, Dr B. Meshram,"Analysis of Signature-based and Behavior-based AntiMalware Approaches", International Journal of Advanced Research in Computer Engineering and Technology, Vol.2, Issue 6, June 2013.

[2] Yong Tang, Bin Xiao and Xicheng Lu, "Signature Tree Generation for Polymorphic Worms", IEEE Transactions on Computers, VOL. 60, NO. 4, APRIL 2011.

[3] Wei Yu, Nan Zhang, Xinwen Fu and Wei Zhao, "Self-Disciplinary Worms and Countermeasures: Modeling and Analysis", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 21, NO. 10, OCTOBER 2010.

[4] Asaf Shabtai, Eitan Menahem and Yuval Elovici, " F-Sign: Automatic, Function-Based Signature Generation for Malware", IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS –PART C: APPLICATIONS AND REVIEWS, VOL. 41, NO. 4, JULY 2011.

[5] REITER, M., AND YEN, T. Traffic aggregation for malware detection. In Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA) (2008).

[6] Wei Yu, Nan Zhang, Xinwen Fu and Wei Zhao, "Self Disciplinary Worms and Countermeasures: Modeling and Analysis", IEEE Transactions on Parallel and Distributed Systems, Vol. 21, NO. 10, OCTOBER 2010.

[7] Shih-Yao Dai, Sy-Yen Kuo. MAPMon: A Host-Based Malware Detection Tool. The 13th IEEE International Symposium on Pacific Rim Dependable Computing. 2007.

[8] Sulaiman A, Ramamoorthy K, Mukkamala S et al. Disassembled code analyzer for malware. Information Reuse and Integration, Conf, 2005 IEEE International Conference on 2005, 398-403.

[9] Ranveer, S., & Hiray, S. SVM Based Effective Malware Detection System. In: 2015 International Journal of Computer Science and Information Technologies, Vol. 6 (4), 2015, 3361-3365.

[10] Khammas, Ban Mohammed, et al. "FEATURE SELECTION AND MACHINE LEARNING CLASSIFICATION FOR MALWARE DETECTION." Jurnal Teknologi 77.1 (2015).

[11] Lu, Yi-Bin, Shu-Chang Din, Chao-Fu Zheng, and Bai Jian Gao. "Using multi-feature and classifier ensembles to improve malware detection." Journal of CCIT 39, no. 2 (2010): 57-72.

[12] G. McGraw and G. Morrisett, "Attacking malicious code: report to the Infosec research council," IEEE Software, 17(5):33 - 41, Sept./Oct. 2000.

[13] S. M. Tabish, M. Z. Shafiq and M. Farooq, "Malware Detection using Statistical Analysis of Byte-Level File Content," CSI-KDD'09, pp.23-31, June 2009.

[14] Vinod, P., R. Jaipur, V. Laxmi, and M. Gaur. "Survey on malware detection methods." In Proceedings of the 3rd Hackers' Workshop on Computer and Internet Security (IITKHACK'09), pp. 74-79. 2009.