Using Deep Learning Model for Network Scanning Detection

Hung Nguyen Viet
Department of Information Security
Le Quy Don Technical University
Ha noi, Viet Nam
hungnv@mta.edu.vn

Quan Nguyen Van
Department of Information Security
Le Quy Don Technical University
Ha noi, Viet Nam
nguyenvanquan87@mail.ru

Linh Le Thi Trang
Department of Intelligent Information Systems
Moscow Institute of Physics and Technology
Moscow, Russia
tranglinh2011@gmail.com

Shone Nathan
Department of Computer Science
Liverpool John Moores University
Liverpool, UK
n.shone@ljmu.ac.uk

ABSTRACT

In recent years, new and devastating cyber attacks amplify the need for robust cybersecurity practices. Preventing novel cyber attacks requires the invention of Intrusion Detection Systems (IDSs), which can identify previously unseen attacks. Many researchers have attempted to produce anomaly - based IDSs, however they are not yet able to detect malicious network traffic consistently enough to warrant implementation in real networks. Obviously, it remains a challenge for the security community to produce IDSs that are suitable for implementation in the real world. In this paper, we propose a new approach using a Deep Belief Network with a combination of supervised and unsupervised machine learning methods for port scanning attacks detection - the task of probing enterprise networks or Internet wide services, searching for vulnerabilities or ways to infiltrate IT assets. Our proposed approach will be tested with network security datasets and compared with previously existing methods.

CCS Concepts

Security and privacy → Intrusion/anomaly detection and malware mitigation → Intrusion detection systems → Artificial immune systems

Keywords

Intrusion Detection, network scanning attacks, Deep Belief Network;

ICFET '18, June 25–27, 2018, Moscow, Russian Federation
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6472-0/18/06…$15.00
https://doi.org/10.1145/3233347.3233379

1. INTRODUCTION

Due to the large usage of computers and computer networks worldwide, computer networks security has become an international concern. Through various scanning techniques, an attacker will attempt to gain information about network configurations, server implementations, and potential vulnerabilities before launching more invasive exploits. Scanning techniques can be categorized into three main groups, namely, TCP, UDP, ICMP. The received response indicates port status and can be helpful in determining a hosts' operating system and other information relevant for launching a future attack. Elias in [3] presented a taxonomy on distributed cyber scanning detection techniques. The approaches are split into four categories, namely, statistical, algorithmic, mathematical and heuristical:

Statistical Approaches: These distributed cyber scanning detection approaches include techniques such as statistical characterization (features) of data samples, extrapolation or interpolation of data based on some best-fit, error estimates of observations, or spectral analysis of a data model.

Algorithmic Approaches: These distributed cyber scanning detection approaches employ step-by-step procedures for calculations, data processing, and formal automated reasoning.

Mathematical Approaches: These distributed cyber scanning detection approaches utilize mathematical models, finite state machines and other algebraic and geometric techniques to achieve their detection task.

Heursitical Approaches: These distributed cyber scanning detection approaches utilize non-formal expert based analysis including, but not limited to, visualization techniques, filter-based heuristics, previous incident analysis, and multidisciplinary techniques.

In recent times, many research groups around the world have applied artificial intelligence models and intelligent computing to IDSs, including network-scanning attacks detection [1,2,5]. In this paper, we will propose a method using Deep Belief Networks to process network data in order to detect the signatures of port scanning attacks. With the ability to efficiently analyze larger multidimensional data, we will be experimenting with the NSL-

KDD and UNSW-NB15 datasets. Our proposed method outperforms previous machine learning methods.

The rest of the paper is organized as follows. In Section II, we overview some port scanning techniques and recent studies for port scanning detection. In Section III, we explain our proposed method using Deep Belief Networks for network scanning attack detection. Experiments and results are shown in Section IV. In Section V, we conclude the paper and present our future work.

2. OVERVIEW OF NETWORK SCANNING ATTACKS AND METHODS FOR DETECTION

In this section we will introduce common network scanning techniques, presenting the principles of operation together with the strengths and weaknesses of each technique. Finally, we will provide a summary of some of the scanning techniques that have been investigated and used recently.

2.1 Port Scanning Techniques

1) Open Scan: Open Scan is the simples scanning technique [3]. This technique utilizes the TCP protocol and the SYN flag to detect TCP ports. When a closed port is targeted, the victim replies with a RST flag. On the other hand, when an open port is detected, the victim replies with an ACK flag. An advantage of this technique is that it can achieve its scan in a very simplistic way without requiring any other functionalities or privileges. This simple technique is easily detected by a firewall.

2) Half-Open Scan: The Half-Open scan, commonly dubbed as the TCP SYN scan, is a common method for port identification that allows the scanner to gather information about open ports without completing the TCP handshake process. Since this scan technique never actually creates a TCP session, it is advantageous in two ways. First, it is not logged by destination applications. Second, it is less stressful to the application service because it does not force the application to initialize or for systems resources to be allocated. On the other hand, this method suffers from one disadvantage. Since there is a need to create new raw packets that do not completely abide by the TCP handshake, the half-open connection process requires some elevated systems privilege [5].

3) Stealth Scans: The aforementioned cyber scanning techniques only use the typical SYN flag to investigate open ports. Hence, they are easily detected and logged by IDSs. These techniques try to avoid filtering devices by employing certain sets of flags other than SYN to appear as legitimate traffic. All these techniques resort to inverse mapping to determine open ports. They are SYN— ACK Scan, IDLE Scan, Fin, Xmas Tree and Null Scans, ACK Scan, Windows Scan and TCP Fragmentation Scan. [5].

4) Sweep Scan: Sweep scans, which do not aim to identify active ports but rather identify active hosts. They are characterized as performing sweeps, since their purpose is to identify the status of as many hosts as possible instead of focusing on an individual host. They operate by generating any request that would prompt a remote stations response. They can be defined as cyber scanning facilitators because they pinpoint active hosts just before the actual scanning techniques of active hosts take place. They are ICMP Echo Request Scan, ICMP Timestamp v Address Mask Scans and TCP SYN Scan.[5]

5) Miscellaneous Scans: cyber scanning insights by shedding light on scans that deal with various protocols. These include the FTP bounce, UDP, IP protocol and RPC scans.

We have a detailed discussion of common network scanning techniques. These attack techniques are most commonly found in network security datasets that are introduced and used to detect network scanning attacks in this paper.

2.2 Network Scanning Detection Methods

In this section we will present some of the scanning detection techniques that have been proposed recently by some research groups. M. Vidhya in [1] used a feature extraction technique and a Support Vector Machine (SVM) to classify scan attacks. The author proposes a feature extraction method for the KDD99 dataset using the Consistency Sybset Evaluation algorithm and the Best First method. The test results show that using SVM with finite datasets is more accurate with the original dataset. The authors achieved a fairly high error of 99.9185 % [1]. However, with the asymmetric data, other research groups often use the measure associated with the attack data.

Authors in [2] present an IDS based on the Support Vector Machine (SVM). It uses evolutionary algorithms to optimize SVM parameters to improve accuracy during intrusion detection.

The author of [4] has provided an AOCD (An Adaptive Outlier Based Scope-Based Scalarized Scan Detection Approach) solution for early detection of high-precision network scanning when tested with KDD99 datasets. In this work, they presented a solution for converting network traffic data into a format that filters and classifiers can operate. It is worth noting that the authors selected random samples in the database, thereby identifying the set of features for clustering detection used in early detection of network scanning techniques. The team tested on the real data collected from the Tezpur University Intrusion Detection System (TUIDS), which confirmed that the solution is capable of detecting cyber-attacks highly accurately.[4].

3. DEEP BELIEF NETWORK

In recent times, deep learning models have been applied successfully in many fields such as image recognition, speech processing, natural language processing and big data processing. In deep-learning models, DBN has been extensively researched and applied, performing well with complex identification and classification problems such as 3D object identification [9], speech recognition [10], arial image processing [11].

DBN is modeled on a multilayer neural network model based on the generative model, according to Hinton [12]. From complex structured data, DBN's network layers will generate output data where the correlation between the data of the same class is more clearly expressed. Differentiating the DBN from other neural network models is the two-phase training algorithm: the unrestricted training mode based on the Restricted Boltzmann Machine (RBM), and the refining phase is the process of supervised learning. RBM is a neural network with a hidden layer and a layer represented in Figure 1.

The RBM, inherited from the Hopfield network model, uses the energy function to evaluate the configuration of the network, including the linkage between neurons, the neuronal freedom coefficient, the state of the neurons (initial value out of neurons) [13]. However, RBMs differ from Hopfield without the links between neurons in the same class, only having no direction between the hidden neurons and the neurons of the class. The energy function of the network in terms of parameters θ determined to be represented by equation 1.

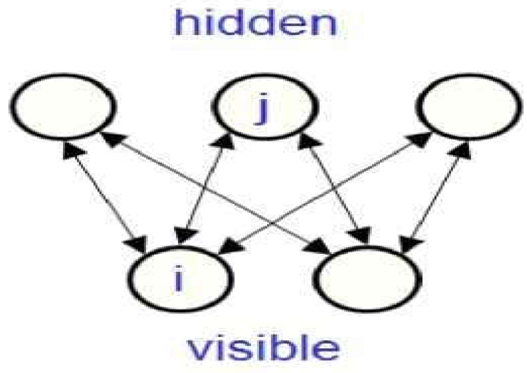


Figure 1: Restricted Boltzmann Machine (RBM)

$$E(v,h|\theta\ =\ -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}\ v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j\, h_j \quad (1)$$

where, θ = (**w**,**b**,**a**) and w_{ij} demonstrating the relationship between the neurons in the visible layer *i* and hidden layer j, *(*b_i*,* a_i is the coefficient of freedom of these two neurons); V and H are the total number of neurons in the visible layer and the hidden layer. Through the energy function, the state probability distribution of the network is represented by the equation 2.

$$p(v|\ \theta) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}} \quad (2)$$

In case the network configuration is known *θ*, the probability of the visible layer *v* is:

$$p(v|\theta) = \frac{\sum_h e^{-E(v,h)}}{\sum_u \sum_h e^{-E(v,h)}} \quad (3)$$

Considering the case that the RBM is binary, the values *v*, *h* only receive 0 or 1. Then, the probability that i-neuron of visible layer with activation state in the case that states of hidden layers are known *h* is determined by the equation 4.

$$p(v_i = 1|h,\theta) = \frac{e^{-E(v_i=1,h)}}{\sum_{v_i}\sum_h e^{-E(v_i,h)}} \quad (4)$$

v_i only receive 0 or 1 so from formula *E*(*v, h*) we have:

$$p(v_i = 1|h,\theta) =$$

$$= \frac{e^{(\sum_{j=1}^{H} w_{ij}h_j + b_i + \sum_{j=1}^{H} a_i h_j)}}{e^{(\sum_{j=1}^{H} w_{ij}h_j + b_i + \sum_{j=1}^{H} a_i h_j)} + e^{(\sum_{j=1}^{H} a_i h_j)}} \quad (5)$$

Or

$$p(v_i = 1|h,\theta) = \sigma\,(b_i + \sum_{j=1}^{H} w_{ij} h_j) \quad (6)$$

Here:

$$\sigma(x) = (1 + e^{-x})^{-1}$$

According to Hinton [13], the network will be trained by updating w_{ij} with the formula (7).

$$\Delta w_{ij} = \varepsilon(\{v_i h_j\}_{data} - \{v_i h_j\}_{recontruction}) \quad (7)$$

Table I: Scanning attacks data in NSL-KDD

Type of attacks	Training	Testing
IPSweep	3599	141
Nmap	1493	73
PortSweep	2931	157
Satan	3633	735
Total:	11656	1106

4. EXPERIMENTS AND RESULTS

4.1 Description of using datasets.

4.1.1 Dataset NSL-KDD:

NSL-KDD is a widely used dataset of current IDSs [14]. The NSL-KDD is an improved dataset from the KDD99 dataset, which eliminates duplicate data, redundant data, training data and more experimental data. It has 41 features as in KDD99, collected and labeled from various test attacks. There are four types of attacks in the form of network scanning attacks that are labeled corresponding to attack techniques or tools that use scan detection *nmap, ipsweep, portsweep* and *satan* with the numbers are described in the Table I.

4.1.2 Dataset UNSW-NB15:

The KDD98, KDD99, NSL-KDD datasets have been in use for a long time, so many of the attack techniques used to generate data are obsolete, without updating existing attack techniques. The UNSW-NB15 dataset [15] has overcome this and is being used extensively in intrusion detection studies. This dataset was released in 2015 at the Cyber Range Lab in New South Wales. The attack data is generated automatically from the IXIA PerfectStorm system with many new attack techniques. Packets are captured, preprocessed and extracted into 49 features. There are a total of 175,341 instances in the training dataset and 82,332 instances in the test dataset. The network scanning attack dataset in UNSW-NB15 is performed by various techniques as described in Section II of this paper. In the training dataset, there are 10491 instances, while in the test dataset there are 3496 instances.

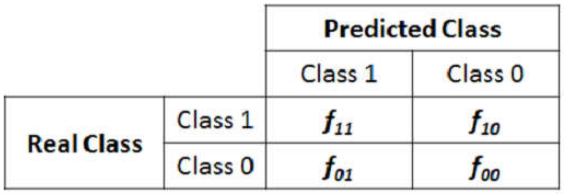
4.2 Experimentation and evaluation of results.

The DBN training program for network scanning attacks detection was built in Python using the TensorFlow library. The experiment was conducted on a 3.6GHz Intel Xeon PC with 16GB of RAM and a NVIDIA GTX 1060 GPU. The final configuration of the DBN was selected through experimental processes, and it has 2 hidden layers, each with 256 nodes. The input vector of the network is dimensioned equal to the numbers of features of the datasets described above.

Table II: Confusion Matrix - NSL-KDD

		DBN-prediction	
		Attacks	Normal
Real data	Attacks	1100	6
	Normal	263	9448

Unsupervised training is done with the greedy algorithm [13], and the refined training phase using the back propagation algorithm. For the NSL-KDD datasets in addition to the described port scan scanners we used unlabeled data "normal" for training (67343 instances) and test (9711 instances). With NSL-KDD, we tested two classifiers: the first is the binary classifier to determine whether the network scan is active. During training and testing, we labelled all probe data as "Probe". The second classifier is trained to classify four types of scanning attacks in NSL-KDD. For the UNSW-NB15 dataset, because the scanning types are labelled together, we only apply a binary classification to determine whether the data is an attack or not. We also used the ''normal" data to train and test the model. The results of the classifiers are often evaluated through the Confusion Matrix and some of the measurements are based on Confusion Matrix. Figure 2 describes how to calculate the confusion matrix for a binary classifier.



		Predicted Class	
		Class 1	Class 0
Real Class	Class 1	f_{11}	f_{10}
	Class 0	f_{01}	f_{00}

Figure 2: Confusion Matrix of binary classifiers

The following measures will be used to evaluate the performance of intrusion detection models.

$$Accuracy = \frac{f_{11}+f_{00}}{f_{11}+f_{10}+f_{01}+f_{00}} \quad (8)$$

$$Error\_rate = \frac{f_{10}+f_{01}}{f_{11}+f_{10}+f_{01}+f_{00}} \quad (9)$$

$$TPR = \frac{f_{11}}{f_{11}+f_{10}} \quad (10)$$

$$FAR = \frac{f_{01}}{f_{01}+f_{00}} \quad (11)$$

$$Precision\ \pi = \frac{f_{11}}{f_{11}+f_{01}} \quad (12)$$

$$Recall\ \rho = \frac{f_{11}}{f_{11}+f_{10}} \quad (13)$$

4.3 Results and evaluation.

Table II is a Confusion Matrix of network layer detection detectors using DBN for the NSL-KDD data. Table III compares the (TPR) and (FAR) of the proposed method with the results of several other methods presented in the articles [1], [4] and [16]. The results showed that the DBN detection rate was higher than other methods, while the false alarm rate was low. Table IV compares Accuracy (acc), Precision (n) and Recall (p) of DBN and other machine learning algorithms [17]. The results show that the DBN archives a more accurate classification of algorithms that have been applied previously. For the UNSW-NB15 datasets, as there are no experimental port-scanning attacks with this dataset, we compare the results of the DBN with two other powerful algorithms for data classification: SVM and RandomForest. Table V shows the results, which indicates that DBN classification results are better than compared algorithms.

Table III. Comparison result of TPR and FAR-NSL-KDD

Method	TPR	FAR
SVM	0.93413	**0.00016**
AOCD	0.98078	0.00667
ANN-MLP	0.82610	0.02811
DBN	**0.99458**	0.02708

5. CONCLUSION

In this paper, we present network scanning techniques and Deep Belief Network with a combination of supervised and unsupervised machine learning model for detecting network scanning attacks. Experiments with the NSL-KDD dataset and the UNSW-NB15 datasets showed that the DBN algorithm had high detection rates for network scanning, while ensuring a lower false alarm rate than the study results before. In the future, we will be experimenting with online data networks to test the algorithm's ability to work in real-time intrusion detection systems.

Table IV. Comparison results of 4 types of scanning attacks detections

Method	(%)	Type of attack			
		ipsweep	nmap	Port sweep	satan
Nave Bayes	acc	93.89			
	π	97.09	97.86	75.54	90.33
	ρ	93.91	93.53	95.86	93.97
k-NN	acc	91.73			
	π	85.74	93.22	84.09	100
	ρ	100	99.89	51.03	77.87
SVM	acc	98.11			
	π	98.35	97.56	96.12	100
	ρ	97.66	100	85.52	99.14
Decision tree	acc	99.50			
	π	99.76	99.77	96	100
	ρ	98.59	100	99.31	99.43
Random Forest	acc	85.06			
	π	80.19	84.29	100	100
	ρ	99.53	99.89	2.07	64.37
MLP with features extraction	acc	99.44			
	π	99.53	99.77	95.92	100
	ρ	98.59	100	97.24	100
DBN	acc	**99.64**			
	π	99.29	100.00	99.36	99.73
	ρ	100.00	98.65	99.36	99.73

Table V: Comparison the results of TPR and FAR UNSW – NB15

Algorithms	TPR	FAR
SVM	99.74	3.20
Random Rorest	99.80	3.31
DBN	**99.86**	**2.76**

6. REFERENCES

[1] Vidhya, M, *Efficient classification of portscan attacks using Support Vector Machine* , Green High Performance Computing (ICGHPC), 2013 IEEE International Conference, 2013

[2] Meijuan Gao, Jingwen Tian, Mingping Xia *Intrusion Detection Method Based on Classify Support Vector Machine* , secondInternational Conference on Intelligent Computation Technology and Automation, ICICTA '09, vol. 2, pp. 391-394.

[3] Elias Bou-Harb, Mourad Debbabi, and Chadi Assi, *Cyber Scanning: A Comprehensive Survey* , IEEE Communications Surveys and Tutorials, 2014, 16.3: 1496-1519.

[4] BHUYAN, Monowar H.; BHATTACHARYYA, Dhruba K.; KALITA, Jugal K, *AOCD: An Adaptive Outlier Based Coordinated Scan Detection Approach* , IJ Network Security, 2012, 14.6: 339-351.

[5] Lee, S. Y, Kim, Y. S., Lee, B. H., Kang, S. H.,Youn, C. H., A *probe detection model using the analysis of the fuzzy cognitive maps*, Computational Science and Its ApplicationsICCSA 2005, 287-291.

[6] Shi Jinn Horng, Ming Yang Su, Yuan Hsin Chen,Tzong Wann Kao, Rong Jian Chen, Jui Lin Lai,and Citra Dwi Perkasa *A novel intrusion detection system based on hierarchical clustering and support vector machines,* Expert Systems with Applications,vol. 38, pp. 306313, January 2011

[7] M H Bhuyan, D K Bhattacharyya, and J K Kalita, *NADO: Network anomaly detection using outlierapproach,* Proceedings of the International Conference on Communication, Computing and Security,pp. 531536. ACM, February 2011.

[8] Webster, A., Gratian, M., Eckenrod, R., Patel, D., Cukier, M. , *An Improved Method for Anomaly-Based Network Scan Detection.* In International Conference on Security and Privacy in Communication Systems (pp. 385-400). Springer, Cham.

[9] Nair, Vinod and Hinton, Geoffrey E., *NADO: 3D object recognition with deep belief nets* Advances in neural information processing systems, 13391347,2009

[10] Abdel-rahman Mohamed, George E. Dahl, and Geoffrey Hinton*Acoustic Modeling using Deep Belief Networks* Audio Speech and Language Processing IEEE Transactions on, vol. 20, pp. 14-22, 2012, ISSN 1558-7916

[11] V. Mnih and G. E. Hinton, *Learning to detect roads in high-resolution aerial images.* in European Conference on Computer Vision., 2010.

[12] Geoffrey E. Hinton, *Learning multiple layers of representation* Trends in cognitive sciences. 11. 428-34. 10.1016/j.tics.2007.09.004.

[13] E Hinton, Geoffrey, Osindero, Simon, Teh, Yee-Whye. *A Fast Learning Algorithm for Deep Belief Nets*. Neural computation. 18. 1527-54. 2016

[14] S. Revathi , Dr. A. Malathi. *A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection*. International Journal of Engineering Research Technology (IJERT). 2. 1848-1853 – 2013

[15] Moustafa, Nour, and Jill Slay. *UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)..* Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[16] Muhammad N., Budi Rahardjo, Riyanto T. Bambang *Improving Performance of Network Scanning Detection Through PCA-Based Feature Selection* International Conference on Information Technology Systems and Innovation (ICITSI) 2014

[17] Ch.Ambedkar, V. Kishore Babu *Detection of Probe Attacks Using Machine Learning Techniques*. International Journal of Research Studies in Computer Science and Engineering (IJRSCSE) , 2015.