International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

Intrusion Detection Systems Vulnerability on Adversarial Examples

Ashutosh Dange, Balaji Chaugule, Pravin Patil
Department of Computer Engineering, Zeal College of Engineering & Research, Pune, Maharashtra, India

Article Info
Volume 8, Issue 2
Page Number : 373-378

Publication Issue :
March-April-2022

Article History
Accepted: 10 March 2022
Published: 22 March 2022

ABSTRACT

Intrusion detection systems define an important and dynamic research area for cybersecurity. The role of Intrusion Detection System within security architecture is to improve a security level by identification of all malicious and also suspicious events that could be observed in computer or network system. One of the more specific research areas related to intrusion detection is anomaly detection. Anomaly-based intrusion detection in networks refers to the problem of finding untypical events in the observed network traffic that do not conform to the expected normal patterns. It is assumed that everything that is untypical/anomalous could be dangerous and related to some security events. To detect anomalies many security systems implements a classification or clustering algorithms. However, recent research proved that machine learning models might misclassify adversarial events, e.g. observations which were created by applying intentionally non-random perturbations to the dataset. Such weakness could increase of false negative rate which implies undetected attacks. This fact can lead to one of the most dangerous vulnerabilities of intrusion detection systems. The goal of the research performed was verification of the anomaly detection systems ability to resist this type of attack. This paper presents the preliminary results of tests taken to investigate existence of attack vector, which can use adversarial examples to conceal a real attack from being detected by intrusion detection systems.

Keywords: Anomaly detection, Adversarial examples, intrusion detection systems.

I. INTRODUCTION

The increasing pace of Internet network develop has a result in an appearance of more complex and difficult to identify threats for computer security. This fact has created a need for prepare automated methods, which can monitor activity in a local computer system or a network and detect intrusion attempts.

One of the possible resolutions for the mentioned problem is Intrusion Detection System (IDS). It is a tool or mechanism which can recognize attack attempt by analyzing the activity of system or network. After a detection IDS can raise the alarm. Every system which

is capable of taking an autonomous decision for further steps, such as connection blocking, is called Intrusion Prevention System (IPS).

Nowadays IDS and IPS became one of the most crucial elements in security infrastructure. As any other, previous method is also an aim of many hackers, who seek to find any weakness, which can compromise used IDS.

This paper presents our attempt to compromised sample IDS based on a neural network by using adversarial examples. To the best of our knowledge there were no other attempts to use adversarial examples to mislead models widely used in anomaly detection.

II. RELATED WORK

After adversarial examples were discovered by Szegedy et al. [3] a large number of researchers have been done for all areas in which machine learning and artificial intelligence have found an application.

The main field of research was image recognition. Kurakin et al. [13] have proved that it is possible to deceive the autonomous vehicles by manipulating a stop sign in a traffic sign recognition system. A similar problem was discussed in [14] where automatic speech recognition was compromised by adversarial commands. Grosse et al. [12] presented how adversarial examples can be used to attack a malware detection system based on neural network. It is the first implementation of generating adversarial examples in cybersecurity and it shows that this method can be used by hackers to hide attempts to attacks. It is also starting point for discussion how to recognize this kind of hacking attacks and make the systems resistant to them.

III. INTRUSION DETECTION SYSTEMS

A. Categories of Intrusion Detection Systems

Intrusion detection systems are a support tool in security infrastructure. They can reduce a cost of maintaining an appropriate security level and convey information about any breach of security.

Two categories of intrusion detection systems can be distinguished:

- Host-based Intrusion Detection Systems (HIDS): they can monitor the system activity on which it has been deployed. HIDS may monitor the integrity of files on a file system, malicious activity on a kernel level and analyze log files for searching a suspicious activity.
- Network-based Intrusion Detection Systems: they focused on monitoring network infrastructure. By analyzing a flow of network packets, inspecting headers and contents it is possible to detect subsequences which can prove that network is an aim of an attack.

Both types of IDS analyze data using one of two strategies:

- Signature-based Intrusion Detection Systems: detection is based on signatures of known attacks and rules defined by an administrator. Such systems can classify known attacks by comparing observed activity with stored patterns, but cannot identify new attacks.
- Anomaly-based Intrusion Detection Systems: they search deviation from normal behaviors. Such situation can be a premise that in monitored system someone is performing an attack. This concept assumes that it is possible to create a model of normal system activity. By using the model and evaluate current measurements it is possible to determine if the observed activity is an anomaly.

B. Methods of anomaly detection.

To perform anomaly detection in the network traffic researchers used algorithms and methods from different classes. First approaches [1] based on statistical point of view. They used statistics to compute a distribution of attributes and apply a statistical inference test to determine if the observed instance is an anomaly.

Soft computing methods represent a heuristic approach which does not provide the exact solution. Despite of that, methods like Neural Networks, Genetic Algorithms or Fuzzy Sets characterize the large degree of flexibility which is crucial for dynamic nature of computer networks.

Within years machine learning algorithms were applied to resolve various problems. Typical usage is image classification, pattern recognition, drug discovery, etc. Based on a training set machine learning methods can be classified into three categories:

- Supervised learning: training set contains labeled exampled and a task is to match a new observation with exactly one class.
- Unsupervised learning: training set does not contain labels or any information about a possible group in it. During training, the algorithm assigns observations to groups and calculate its level of similarity.
- Reinforcement learning: in this problem algorithm perform an action and then received feedback. Information may indicate rewards or punishments. Based on this value algorithm is pitched.

Algorithms from first two groups have been applied in network anomaly detection problem. The assumption is that attacks can be detected because they are very uncommon events and can be classified by model as unlikely to occur.

Efficiency and accuracy of anomaly detection system often describe with confusion matrix (Table I). A precise description of the matrix and metrics which are used was provided by Bhuyan et al. [2].

TABLE I. CONFUSION MATRIX

Original class	Predicted class	
	Positive	*Negative*
Positive	True positive – correct detection	False negative – incorrect rejection
Negative	False positive – false alarm	True negative – correct rejection

Many implementations of anomaly detection report high numbers of false alarm. It is adverse reaction and has an effect of human intervention need. Network traffic is a large dataset and performing manual analyze is usually impossible.

IV. ADVERSARIAL EXAMPLES

A. Adversarial examples description

Research performed by Szegedy et al. [3] disclosed that even a little variation in classified observation might cause misclassification. This error is revealed in a wide variety of models which have been trained on distinct datasets and used different classification algorithms.

Szegedy et al. [4] presented detailed explanations why the wrong classification is made. In general, their concept assumes that linear models are slightly inert which means that they can distinguish examples on a specific level because every model uses a limited number of bits for every feature. This is a constraint in the decision-making process because classifier has to discard differences in feature value which are under a precision level. Szegedy called this “accidental steganography” because for high dimensional problems it is possible to make many infinitesimal changes to the input.

Adversarial training may have result in generating examples that will be classified as any other class than legitimate source class, it is an untargeted attack. The other approach, a targeted attack, is to conform any sample to the selected target class.

B. Adversarial examples against Intrusion Detection Systems

We consider a possibility to take advantage of adversarial examples as a potential attack vector on intrusion detection systems. Models are used by machine learning algorithms to analyze network

traffic are high dimensional even after applied reduction methods such as Principal Component Analysis. This fact may create an opportunity to insert some perturbations in observed flows and mislead classifier.

Many dimensions should allow to modify several feature values which can lead to liken attacks packets flow to a normal communication between two hosts. We expect that this task may be even easier because many network devices such as routers or network cards can correctly interpret packets with invalid header values. If it can be confirmed it may be possible to use the method called fuzzing to simply fabricate a malicious network

traffic which cannot be correctly identified by intrusion detection systems.

V. TESTING ENVIRONMENT

A. NLS KDD Dataset

Within years a few datasets have been used to evaluate network anomaly detection systems. The best-known dataset is KDDcup99. It was prepared by Stolfo et al. [6] as result of participating in The Third International Knowledge Discovery and Data Mining Tools Competition. This dataset consists of approximately 4 900 000 samples in which 300 000 represent 24 attack types. Every observation is described by 41 features and labeled as an attack or normal.

KDDcup99 has been criticized by many researchers, e.g. Tavallaee et al. [7] Revathi and Malathi [8]. They discovered that KDDcup99 contains many redundant records and irregularities such as malicious packets have a TTL of 126 or 253 while normal samples have 127 or 254.

To solve these issues, a new dataset, NSL-KDD [9], is proposed, which consists of selected records of the complete KDD dataset. Main advantages of NSL-KDD are:

- Redundant records have been excluded from the training set to make classifiers not biased.
- Duplicated records have been eliminated to solve the problem with the performance of methods which have better detection rates on the frequent records.
- There is no more need to select a group of observations for a training and testing sets, algorithms can be evaluated on the complete set.

Nowadays it is recommended to stop using the KDDcup99 dataset, the corresponding message has been published in [10]. We determined to use the NSL-KDD dataset to evaluate our approach because it resolves many of issues in the KDDcup99 [9]. There is a lack of public datasets which can represent real networks and NSL-KDD has been used in some research which indicates that it can be used as a benchmark dataset.

Many of features in NLS-KDD are categorical. To solve this problem, we have decided to use the one-hot-code for each of these features. That method increases a dimensionality but number of observations in the dataset is high enough to preserve required classification quality.

For numeric features we apply the z-score normalization because range of values varies widely depending on attribute. Standardization is widely used to avoid the dominant influence of a specific group of features on the classification result.

B. Neural network

In our research, we are using reference neural network. A neural network is state of the art approach for solving a variety of tasks like classification, regression and dimensionality reduction.

Neural Networks mimic a human brain recognition mechanism. They consist of many elements known as neurons, which are connected into layers. Neural networks systematically change the interconnection strengths, or synaptic weights in the process of learning. Each neuron in network layer applies the activation function to produces an output used as an input by the neurons of next layer.

Our architecture includes 3 hidden layers of 100 neurons each. To tune numbers of neurons we use Grid Search algorithm. We search for the optimal number of neurons in values from 1 to 100 in steps of 5. We use the rectified non-linearity as the activation function for each neuron. Last layer in our neural network is softmax layer, which is used to normalize the output of the network to a probability distribution. To train our network, we use standard gradient descent with batches of size 1000 that are split into training and validation sets, using 100 training epochs per iteration. We also define a condition of early stopping. Training procedure stops when value of loss function, which is cross entropy, does not change more than 0.001 in last 5 epochs and usually it is achieved within 50 epochs. We implemented this algorithm to avoid over-fitting.

C. Fast gradient sign method

A crucial point for our tests of anomaly detection system is generating adversarial examples. Goodfellow et al. [4] proposed method called Fast Gradient Sign. It linearizes the cost function around of the point that should be misclassified. It selects a perturbation by differentiating this cost function with respect to the input itself.

The perturbation can be expressed as:

$$\eta = \varepsilon \, \mathrm{sign}(\nabla_x J(\theta, x, y))$$

where x is the input sample, y is the target, ε is the magnitude of the perturbation, θ is the parameters of a model and J(θ, x, y) is a cos function that was used to train the neural network.

In our research, we use L1 norm, which was originally proposed by Grosse et al. [12].

VI. EVALUATION

The main goal of our experiments was to prove that typical input for anomaly detection systems designed for network traffic analyze has sufficient dimensionality to use effectively adversarial examples generation algorithms.

The results of our experiments are presented at Tables II - V. First two of them demonstrate the confusion matrix obtained from our intrusion detection system based on the neural network which has classified test set without adversarial examples and the table with corresponded statistics. Table IV and Table V are accordingly confusion matrix and statistics from this matrix for the same test set, but after applying Fast Gradient Method. As a result all anomaly have been modified to imitate normal network packets.

TABLE II. CONFUSION MATRIX FOR THE TEST SET WITHOUT ADVERSARIAL EXAMPLES

	Predicted class	
Original class	Anomaly	Normal
Anomaly	14644	70
Normal	127	16652

TABLE III. BINARY CLASSIFICATION STATISTICS FOR THE TEST SET WITHOUT ADVERSARIAL EXAMPLES

Sensitivity	0,9952
Specificity	0,9924
Precision	0.9914
Negative predictive value	0.9958
False positive rate	0.0076
False negative rate	0.0048
False discovery rate	0.0086
Accuracy	0.9937

TABLE V. BINARY CLASSIFICATION STATISTICS FOR THE TEST SET WITH ADVERSARIAL EXAMPLES

Sensitivity	0
Specificity	1
Precision	-
Negative predictive value	0.5328
False positive rate	0
False negative rate	1
False discovery rate	-
Accuracy	0.5328

The obtained experiment result indicates that for a reference dataset, which is NLS KDD, it is possible to generate adversarial examples by Fast Gradient Sigh Method, which will lead to complete misclassification of potential network attack. Our intrusion detection system, which based on neural network, has been completely compromised by adversarial examples. False negative rate shows that all harmful examples have been classified as normal network traffic. This fact confirms that adversarial examples generating method originally designed for image recognition can be applied in security area. Constantly growing computer networks and volume of transmitted data require implementation of intelligent IDS. Our research shows that adversarial examples are serious threat for recent solutions and lead to potential abuse.

As Papernote et al. [11] indicted, for image classification adversarial examples lead to around 97% misclassification rate. Our results conduct the proposal that for network traffic datasets is possible to obtain the same or even better result than for image classification.

Our researches are based only on simulation on NLS KDD dataset. In a real application, a number of misclassified network packets will be probably reduced by error correction mechanisms that are implemented in network devices, such as routers and servers. Moreover, we have not considered the case in which monitored network has other defenses mechanisms like firewalls.

VII. FUTURE WORK

Next step in evaluating a real influence of adversarial examples on intrusion detection systems is to build a functioning network and perform a similar experiment like KDD99 Cup. It has to be proved how much of generated network traffic can be correctly recognize by network devices.

Grosse et al. [12] presented defensive mechanisms that reduce a number of misclassified malware examples. These two problems, intrusion detection and malware detection, are similar, which provide an assumption, that the same defensive methods can be successfully applied to the intrusion detection domain.

VIII. REFERENCES

[1]. Smaha, S.E.: Haystack: an intrusion detection system. In: Fourth Aerospace Computer Security Applications Conference, pp. 37-44. IEEE, Orlando, FL, USA (1988).

[2]. Bhuyan, Monowar H., Bhattacharyya, D. K., Kalita, J. K.: Network Anomaly Detection : Methods, Systems and Tools. IEEE Communications Surveys & Tutorial 16(1), 303-336 (2014).

[3]. Szegedy, Ch., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., Fergus, R. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013).

[4]. Goodfellow, I. J., Shlens, J., Szegedy, Ch. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014).

[5]. Butti, L. presentation: Wi-Fi Advanced Fuzzing, http://www.blackhat.com/presentations/bh-europe- 07/Butti/Presentation/bh-eu-07-Butti.pdf , last accessed 2017/05/30

[6]. Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., Chan, P. K. “Cost-Based Modeling for Fraud and Intrusion Detection: Results from the JAM Project,” in Proc. of the DARPA Information Survivability Conference and Exposition, vol. 2. USA: IEEE CS, 2000, pp. 130–144

[7]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. of the 2nd IEEE International Conference on Computational Intelligence for Security and Defense Applications. USA: IEEE Press, 2009, pp. 53–58.

[8]. Revathi, S., and A. Malathi. "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection." (2013).

[9]. NSL-KDD, “NSL-KDD data set for network-based intrusion detection systems,” http://iscx.cs.unb.ca/NSL-KDD/, March 2009.