UDC 004.732

Detection of Unauthorized Actions in Networks Using Wavelet Analysis

Pavlo Hrynchenko[1]

[1] National University "Zaporizhzhia Polytechnic", st. Zhukovsky, 64, Zaporizhzhia, 69063, Ukraine

Abstract

Signal processing techniques are used to analyze and detect network anomalies because of their ability to detect new and unknown intrusions. The paper proposes a method of modeling network signals for the detection of network anomalies, which combines wavelet approximation and the theory of system identification. To characterize the behavior of network traffic, fifteen functions are provided, which are used as input signals within the system. At the same time, it is assumed that security violations within the network can be detected by checking abnormal patterns of system functioning according to audit data.

Despite the fact that machine learning methods have achieved significant results in detecting network anomalies, they still face the difficulty of using the implemented algorithms, in the presence of differences in the behavior of the training data and test data, which in turn leads to inefficient performance of the algorithms. This effect is exacerbated by the limitation of algorithms to detect previously unknown types of attacks due to the large number of false positives.

The paper develops a new method of modeling network signals for detecting anomalies in networks using wavelet analysis. In particular, the general architecture of the approach consists of three components: feature analysis, modeling of normal network traffic based on wavelet approximation and prediction using ARX model, intrusion or non-intrusion decision making

The result is evaluated using the DARPA intrusion detection dataset, which performs a comprehensive analysis of the intrusions in the dataset. Evaluation results show that this approach provides a high level of detection of both instances and types of attacks.

Keywords: Wavelet analysis, network intrusion detection system, network security

Introduction

Traditionally, intrusion detection methods are divided into two categories: abuse detection and anomaly detection. Exploit detection is based on the assumption that most attacks leave a set of signatures in the network packet stream or in audit logs, and therefore attacks can be detected if these signatures can be identified by analyzing audit logs or network traffic behavior. However, exploit detection approaches are strictly limited to the latest known attacks. Detecting new attacks or variants of known attacks is one of the biggest challenges facing exploit detection.

To overcome the disadvantages of detecting misuse, the concept of anomaly detection has been formalized. It has been hypothesized that security breaches can be detected by checking for abnormal patterns of system usage from audit data. As a result, most anomaly detection methods attempt to establish normal activity profiles by computing various metrics, and intrusions are detected when the actual behavior of the system deviates from the normal profiles. According to the characteristics of the monitored sources, anomaly detection can be classified into host-based and network-based. Typically, a host-based anomaly detection system runs on a local monitored host and uses its log files or audit log data as a source of information. The main limitation of host-based anomaly detection is its ability to detect distributed and coordinated attacks that show patterns in network traffic. In contrast, network anomaly detection aims to protect the entire network from intrusion by monitoring network traffic either on designated hosts or on specific sensors, and thus can simultaneously protect a large number of computers with different operating systems from remote attacks such as port scanning, distributed denial of service attacks, distribution of computer worms that pose a serious threat to the current Internet infrastructure. Therefore, in this work, attention is focused on the detection of network anomalies.

Early network anomaly detection systems are self-learning, meaning they automatically form an opinion about what a subject's normal behavior is. Such self-learning methods include statistical model-based early anomaly detection approaches, AI (artificial intelligence)-based approaches, or biological model-based approaches. Although machine learning techniques have so far achieved good results in detecting network anomalies, they still face some serious challenges, including the security of machine learning, behavioral dissimilarity in training and testing data that will completely hinder anomaly detection algorithms, and limited ability to detect previously unknown attacks due to a large number of false alarms.

Considered as an alternative to traditional network anomaly detection approaches or data preprocessing for conventional detection approaches, signal processing techniques have recently been successfully applied to network anomaly detection due to their ability to detect point change and transform data (e.g., with the CUSUM DdoS attack detection algorithm).

This paper proposes a new network signal modeling technique for network anomaly detection. General architecture of our approach, consists of three components, namely, feature analysis, normal network traffic modeling based on wavelet approximation and prediction using the ARX model, and intrusion decision.

During feature analysis, we identify and generate fifteen features to characterize network traffic behavior, and we expect that the greater the number of features, the more accurately the traffic volume information for the entire network will be characterized. This differs from current wavelet-based network anomaly detection approaches, as most of them use a limited number of features (i.e., the number of packets per time slot) or existing features from a public intrusion detection dataset as input signals. Based on the proposed fifteen characteristics, normal daily traffic is then modeled and represented by a set of wavelet approximation coefficients that can be predicted by the ARX model [1]. Compared with current approaches that try to extract different frequency components from existing network signals, our approach is more general and adaptive because the ARX model used to predict the expected value of frequency components is trained from network traffic data collected in the current deployment network. The output for the normal daily traffic model is a residual value that represents the deviation of the current input signal from normal/regular behavioral signals. The residuals are finally fed into the intrusion decision engine, which runs the outlier detection algorithm and makes the intrusion decision.

During feature analysis, we identify and generate fifteen features to characterize network traffic behavior, and we expect that the greater the number of features, the more accurately the traffic volume information for the entire network will be characterized. This differs from current wavelet-based network anomaly detection approaches, as most of them use a limited number of features (i.e., the number of packets per time slot) or existing features from a public intrusion detection dataset as input signals. Based on the proposed fifteen characteristics, normal daily traffic is then modeled and represented by a set of wavelet approximation coefficients that can be predicted by the ARX model [1]. Compared with current approaches that try to extract different frequency components from existing network signals, our approach is more general and adaptive because the ARX model used to predict the expected value of frequency components is trained from network traffic data collected in the current deployment network. The output for the normal daily traffic model is a residual value that represents the deviation of the current input signal from normal/regular behavioral signals. The residuals are finally fed into the intrusion decision engine, which runs the outlier detection algorithm and makes the intrusion decision.

1. Related Works

Today's development of computer networks affects most spheres of economic activity. A significant number of enterprises and organizations around the world use computer networks to manage production processes and personnel, allocate resources etc. This gives them a number of significant advantages - speeding up production processes, increasing mobility and speed of access to information and services, the possibility of remote management of banking invoices, ordering and paying for goods and services. This led to a significant increase in the value of information circulating in computer networks.

Ensuring the operability of networks, as well as the operability of the information systems operating in them, depends not only on the reliability of the equipment used, but also on the ability of the network to resist targeted actions aimed at disrupting its operation.

It should be noted that attacks on information systems are becoming more sophisticated, larger and more intense every year. Therefore, the issue of improving intrusion detection systems, the main task of which is the detection of network attacks, attempts at unauthorized access to the network and the use of its resources, is becoming more and more urgent.

Over the last several decades, computer networks have become a global phenomenon, the development of which affects most spheres of economic activity. Robert Metcalfe, who participated in the creation of Ethernet, was one of the first to quantify the importance of networks: according to the assessment, the "importance" of a network is in all senses proportional to the square of the number of nodes in it. That is, dependence on the normal operation of networks is growing faster than the networks themselves. Ensuring the performance of the network and the functioning of information systems in it depends not only on the reliability of the equipment, but also, most often, on the ability of the network to resist targeted actions aimed at disrupting its operation [3].

Currently, the creation of systems guaranteed to be resistant to harmful influences and computer attacks is associated with significant costs of both time and material resources. In addition, there is a well-known inverse relationship between the ease of use of the system and its security: the stronger the protection system, the more difficult it is to use the main functionality of the information system.

General-purpose operating systems are used to organize information systems. Due to the high complexity and high cost of the development of protected systems, for which the feasibility of the main security theorem would be formally proved - the failure to remove the system from a safe state for any sequence of actions of interacting objects (which requires use), the direction of information security began to actively develop, related to the detection (and subsequent response) of security violations of information systems, which allows to obtain an effective solution to the issue of system security and provides an opportunity to close vulnerabilities in the security of systems until their correction. This direction was named "detection of attacks" (intrusion detection). Over the past years, hundreds of attack detection systems have been created as part of academic developments for various platforms: from mainframe systems to modern general purpose operating systems, DBMS and common applications [4].

The creation of effective systems for the protection of information systems is also faced with a lack of computing power. The development of computer networks is subject to two trends, called Moore's Law and Gilder's Law. Moore's law speaks of the annual doubling of the productivity of computers available for the same cost, and Gilder's law - of the tripling of the bandwidth of communication channels over the same period. Thus, the growth of the computing power of network nodes lags behind the growth of the volume of information transmitted over the network, which every year increases the requirements for the computational complexity of the algorithms of information protection systems.

2. Methodology

The approach consists of three components, namely, feature analysis, wavelet approximation and ARX-based modeling of normal daily traffic, and intrusion decision. In this section, we will discuss each component in detail.

2.1. Analysis of functions

The main goal of feature analysis is to select and highlight reliable network features that can distinguish anomalous behavior from normal network activity. Since most modern network intrusion detection systems use network flow data (e.g., netflow, sflow, ipfix) as information sources, we focus on features from a flow perspective.

The following five key metrics are used to measure the behavior of the entire network:

FlowCount. A flow consists of a group of packets traveling from a specific source to a specific destination during a specific period of time. Currently, there are various flow definitions, such as netflow, sflow, ipfix. Essentially, a single network flow must contain a source (consisting of source IP address, source port), destination (consisting of destination IP address, destination port), IP protocol, number of bytes, number of packets. Threads are often thought of as sessions between users and services. Because attack behavior is usually different from normal user activity, it can be detected by observing flow characteristics.

AverageFlowPacketCount. The average number of packets in the stream per time interval. Most attacks occur with an increase in the number of packets. For example, distributed denial of service (DDoS) attacks often generate a large number of packets in a short time to quickly consume available resources.

AverageFlowByteCount. The average number of bytes in the stream per time interval. With this metric, we can determine whether network traffic consists of large packets or not. Some previous denial-of-service (DoS) attacks use the maximum packet size to consume computing resources or overload data paths, such as the ping of death (pod) attack [5].

AveragePacketSize. The average number of bytes per packet in the stream during the time interval. It describes the size of packets in more detail than the AverageFlowByteCount function above.

FlowBehavior. Ratio of FlowCount to AveragePacketSize. It measures the abnormality of flow behavior. The higher the value of this ratio, the more abnormal the flows, since most probing or surveillance attacks launch a large number of small-packet connections to achieve maximum probing performance.

Table 1
List of functions

Notation of features	Description
f_1	Number of TCP flows per minute
f_2	Number of UDP streams per minute
f_3	Number of ICMP flows per minute
f_4	Average number of TCP packets per flow for 1 minute
f_5	Average number of UDP packets per flow during 1 minute
f_6	Average number of ICMP packets per flow during 1 minute
f_7	Average number of bytes per TCP stream for 1 minute
f_8	Average number of bytes per UDP stream during 1 minute
f_9	Average number of bytes per ICMP flow during 1 minute
f_{10}	Average number of bytes per TCP packet in 1 minute
f_{11}	Average number of bytes per UDP packet in 1 minute
f_{12}	Average number of bytes per ICMP packet in 1 minute
f_{13}	Ratio of number of flows to bytes per packet (TCP) during 1 minute
f_{14}	Ratio of number of flows to bytes per packet (UDP) during 1 minute
f_{15}	Ratio of number of flows to bytes per packet (ICMP) during 1 minute

Based on the above five metrics, we define a set of characteristics to describe network-wide traffic information. We use the 15-dimensional feature vector $f \in F, \{f_i\}_{i=1,2,\ldots,15}$, which is listed in Table 1.

Empirical observations of network traffic flow logs show that network traffic volumes can be characterized and distinguished using these features.

2.2. Modeling normal network traffic using Wavelet and ARX

This section briefly reviews the basic theoretical concepts of wavelet transform and system identification, and then provides information on how to model typical daily network traffic signals in the proposed approach.

2.2.1. Third level heading

The Fourier transform is only good for studying stationary signals, where all frequencies are assumed to exist at all times, and is not sufficient for detecting compact patterns. To solve this problem, the short-time Fourier transform (STFT) was proposed, in which Gabor localized the Fourier analysis by considering a sliding window. The main limitation of STFT is that it can provide good resolution in frequency or in time (depending on the window width).

To have a coherence time proportional to the period, Morlet proposed a wavelet transform that can achieve good frequency resolution at low frequencies and good time resolution at high frequencies. Discrete wavelet transform (DWT) is used in the work, since the network signals we are considering have a cutoff frequency. DWT is a multi-step algorithm that uses two basic functions called the wavelet function $\psi(t)$ and the scaling function φ(t) to dilate and shift the signals. Two functions are then applied to transform the input signals into a set of approximation coefficients and detail coefficients with which the input signal X can be reconstructed [6, 1].

System identification refers to the problem of identifying mathematical models of dynamic systems using observed data from the system. In a dynamic system, its output depends on both the input and the previous results. As we know, the ARX model is widely used for system identification. Let x(t) represent the input to the regressor or predictor, and let y(t) denote the output produced by the system we are trying to model. Then ARX [p, q, r] can be represented by the following linear difference equation:

$$y(t) = \sum_{i=1}^{p} a_i y(t-i) + \sum_{i=r}^{q} b_i x(t-i) + e(t), \qquad (1)$$

where a_i and b_i are model parameters. Given an ARX model with parameters θ, we have the following equation to predict the value of the following outcome:

$$\hat{y}(t|\theta) = \sum_{i=1}^{p} a_i y(t-i) + \sum_{i=r}^{q} b_i x(t-i), \qquad (2)$$

and prediction error:

$$\varepsilon(t) = y(t) - \hat{y}(t|\theta), \qquad (3)$$

The goal of determining a particular set of parameter values from a given parameter space is to minimize the prediction error. The method of least squares estimation is usually used to obtain the optimal value of θ parameters.

2.2.2. Simulation of normal network traffic

Modeling normal network traffic consists of two steps, namely wavelet decomposition/reconstruction and autoregressive model generation. As a rule, the implementation of the wavelet transform is based on a bank of filters or a pyramidal algorithm. In practical implementation, signals are passed through a low-pass filter (H) and a high-pass filter (G) at each stage. Given a signal of length l, we expect to obtain a filtered signal of length l. Since there are two filters in each filtering stage, the total number of filtered signals is 2l. To eliminate redundancies in the signals, we can downsample the filtered low-pass and high-pass signals by half without losing information. The amount of data can be reduced by downsampling, since in this case we are only interested in approximations. After the low-level details are filtered out, the remaining coefficients represent a high-level summary of the signal's behavior, and we can therefore use them to establish a signal profile that characterizes the expected behavior of network traffic throughout the day [7, 3].

Although there are also some other algorithms, such as trous and redundant wavelet transforms, which do not downsample signals after filtering, we use the filterbank algorithm in simulating normal network traffic. Therefore, during the wavelet decomposition/reconstruction process, the original signals are transformed into a set of wavelet approximation coefficients that represent the approximate summary information about the signal, since the details have been removed during filtering.

Next, to estimate the ARX parameters and build the ARX prediction model, we use the wavelet coefficients from one part of the training data as input and the wavelet coefficients from the other part of the training data as the model fitting data. The ARX fitting process is used to estimate the optimal parameters based on least squares errors.

Once we have a prediction model for normal network traffic, we can use it to distinguish abnormal signals from normal ones. When the model inputs include only normal traffic, its outputs, called residuals, will be close to 0, meaning that the predicted value produced by the model is close to the actual normal input. Otherwise, when the input to the model includes normal and abnormal traffic, the residuals will include many peaks where anomalies occur. In this case, residuals are treated as a kind of mathematical transformation that tries to zero out normal network data and amplify abnormal data.

2.3. Emission Detection and Intrusion Decisions

According to the above section, we assume that the higher the value of the residuals, the more anomalous the flow. As a result, to identify residual peaks (or outliers) [8], we implement an outlier detection algorithm based on a Gaussian mixture model (GMM) and make an intrusion decision based on the results of the outlier detection algorithm.

In pattern recognition, it has been found that a Gaussian mixture distribution can approximate any distribution with arbitrary accuracy if a sufficient number of components are used, and thus an unknown probability density function can be expressed as a weighted finite sum Gaussian with different parameters and mixing proportions. Given a random variable x, its probability density function p(x) can be represented as a weighted sum of components:

$$p(x) = \sum_{i=1}^{k} a_i f_i(x; \mu_i . v_i), \tag{4}$$

where k is the number of components of the mixture; α_i $(1 \leq i \leq k)$ denotes mixing proportions that always sum to 1. $f_i(x;\ \mu_i,\ \nu_i)$ refers to the component density function, where μ_i denotes the mean of the variable x and ν_i is the variance of x. The density function can be a multivariate or univariate Gaussian distribution.

The expectation-maximization (EM) algorithm has been proposed as an efficient algorithm for GMM parameter estimation. Assume that the mixture component is a one-dimensional Gaussian EM algorithm for GMM can be described as follows:

1. Initialization of a set of parameters $\theta^0 = \langle a_i^0, \mu_i^0, \sigma_i^0 \rangle$
2. E-step: for each given X∼{xn | n = 1, 2, ... , N} and for each component of the mixture k calculate the posterior probability p(i | xn) by solving the equation:

$$p(i|x_n) = \frac{a_i N(x_n; \mu_i, \sigma_i)}{\sum_{i=1}^{k} a_i N(x_n; \mu_i, \sigma_i)}. \tag{5}$$

3. M-step: re-estimation of parameters based on posterior probabilities p(i | xn)

$$p(i|x_n) = a_{inew} = \frac{1}{N} \sum_{n=1}^{N} p(i|x_n)\ , \tag{6}$$

$$\mu_{i\text{new}} = \sum_{n=1}^{N} \left(\frac{p(i|x_n)}{\sum_{n=1}^{N} p(i|x_n)}\right) x_n\ , \tag{7}$$

$$\sigma_{i\text{new}} = \sum_{n=1}^{N} \left(\frac{p(i|x_n)}{\sum_{n=1}^{N} p(i|x_n)}\right). \tag{8}$$

4. Moving to step 2, the algorithm will not converge.

At the E-step (waiting step) of the above EM algorithm, the posterior probability $p(i \mid x_n)$ is calculated for each given $X\sim\{x_n \mid n = 1, 2, ... , N\}$ and each component of the mixture $i(1 \leq i \leq k)$. At the M-step (maximization step), the set of parameters {αi, μi, νi} is reestimated based on the posterior probabilities $p(i \mid x_n)$ that maximize the likelihood function. The EM algorithm starts with some initial random parameters and then repeatedly applies E-step and M-step to obtain better parameter estimates until the algorithm converges to a local maximum.

The outlier detection algorithm is based on the posterior probability generated by the EM algorithm [9]. The posterior probability describes the probability that the data pattern approximates a specified Gaussian component. The higher the posterior probability that the data pattern belongs to a particular Gaussian component, the better the approximation. As a result, the data are assigned to the corresponding Gaussian components according to their posterior probabilities. However, in some cases there are patterns in the data such that the posterior probability of belonging to any GMM component is very low or close to zero. These data are naturally treated as outliers or noisy data. The thresholds correspond to the termination conditions associated with the outlier detection algorithm: the first one measures the absolute accuracy required by the algorithm, and the second one is the maximum number of iterations of our algorithm. The emission threshold value refers to the minimum mixing ratio. Once the mixing proportion corresponding to one defined Gaussian component is below the outlier threshold, the posterior probability that the data pattern belongs to that Gaussian component will be set to 0.

The intrusion decision-making strategy is based on the outlier detection results: if no outliers are detected, the network flows are normal; otherwise, the network flows represented by this emission are marked as intrusions.

Conclusions

The paper proposes an approach to the detection of network anomalies based on the wavelet transform and the theory of system identification. The input signal is a 15-dimensional feature vector, which is defined to characterize the behavior of network flows. A prediction model for normal traffic is introduced, in which the wavelet coefficients play an important role because they are used as external inputs to the ARX model, which predicts the signal approximation coefficient. The traffic prediction model output measures the difference between normal and abnormal activity. Empirical observations show that the peaks of the residuals always correspond to the locations where the attacks occur. A GMM-based outlier detection algorithm is implemented to detect peaks from a set of residuals. Decisions are made based on the results of the proposed emission detection algorithm.

Discrete wavelet transformation is used in the work, since the network signals under consideration have a cutoff frequency, the basis functions of which are used to transform the input signals into a set of approximation coefficients and detail coefficients, which can be used to reconstruct the input signal. Modeling of normal network traffic consists of two stages - wavelet decomposition/reconstruction and autoregression model generation. In practical implementation, signals pass through low- and high-pass filters at each stage. The size of the data can be reduced by downsampling, since in this case only approximate values are of interest. After the low-level details have been filtered out, the other coefficients are a high-level summary of the signal behavior, and thus can be used to create a signal profile that characterizes the expected behavior of network traffic. In the process of wavelet decomposition/reconstruction, the original signals are transformed into a set of wavelet approximation coefficients, which represent an approximate summation of the signal, since details are removed during filtering. To estimate the ARX parameters and generate a prediction model, the wavelet coefficients of different parts of the training data are used as input and model fitting data. The ARX fitting process is used to estimate the optimal parameters based on the least squares method.

Once a predictive model for normal network traffic is obtained, it can be used to identify abnormal signals from normal ones. When the model inputs include only normal traffic, its outputs, called residuals, will be close to 0, meaning that the predicted value generated by the model is close to the actual normal behavior input. Otherwise, when the input to the model includes normal traffic and abnormal traffic, the residuals will include many peaks where anomalies occur. The residuals are fed into the intrusion decision-making engine, which runs an outlier detection algorithm that makes a decision about a possible intrusion.

References

[1] J. Ryan, Intrusion detection with neural networks. Advances in neural information processing systems / J. Ryan. – Morgan Kaufmann Publishers, 2002. – 989 p.

[2] P. Kukielka, Analysis of different architectures of neural networks for application in intrusion detection systems. International Multiconference on Computer Science and Information Technology / P. Kukielka. – IMCSIT, 2008. – 811 p.

[3] Y. Huang, F. Zhou, J. Gilles, “Empirical curvelet based Fully Convolutional Network for supervised texture image segmentation”, Neurocomputing, Vol. 349, 31–43, 2019.

[4] B. Hurat, Z. Alvarado, and J. Gilles. “The Empirical Watershed Wavelet,” Journal of Imaging, Special Issue 2020 Selected Papers from Journal of Imaging Editorial Board Members,” Vol. 6, No. 12, 140, 2020.

[5] A. R. Adly, Critical aspects on wavelet transforms based fault identification procedures in HV transmission line. IET Gener. Transm. Distrib. 2016, 10, 508–517.

[6] A. Alshawawreh, Wavelet transform for single phase fault detection in noisy environment. In Proceedings of the 2014 IEEE 8th International Power Engineering and Optimization Conference (PEOCO2014), Langkawi, Malaysia, 24–25 March 2014; pp. 429–434.

[7] S. Nathan, T. Ngoc, "A deep learning approach to network intrusion detection." IEEE Transactions on Emerging Topics in Computational Intelligence, 2 (1) (2018), pp. 41-50

[8] Osken, Sinem Intrusion Detection Systems with Deep Learning: A Systematic Mapping Study. 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), 1-4. IEEE.

[9] R. Vinayakumar, Applying convolutional neural network for network intrusion detection. 2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI), 1222-1228. IEEE.