Using Random Forests for Network-based Anomaly detection at Active routers

Conference Paper · February 2008

DOI: 10.1109/ICSCN.2008.4447167 · Source: IEEE Xplore

4 authors, including:

Jayashree Padmanabhan
Anna University, Chennai
60 PUBLICATIONS **395** CITATIONS
SEE PROFILE

N. Srinivasan
11 PUBLICATIONS **79** CITATIONS
SEE PROFILE

IEEE-International Conference on Signal processing, Communications and Networking
Madras Institute of Technology, Anna University Chennai India, Jan 4-6, 2008. Pp93-96

Using Random Forests for Network-based Anomaly detection at Active routers

G.Prashanth, [1]V.Prashanth, P.Jayashree, N.Srinivasan
Department of Information Technology, MIT Campus,Anna University.
[1]prash_49@yahoo.com

***Abstract*:Network Intrusion Detection Systems (NIDSs) is one of the primary components in any network security infrastructure. Currently, many NIDSs that are developed are rule-based systems and their performances mainly depend on the rule sets. But rules formation becomes a tedious and time consuming task due to the enormous amount of network traffic. This is overcome by using data mining techniques. These techniques build intrusion detection models adaptively. Random forests is an efficient data mining algorithm which can be used for network intrusion detection This can be used for real time intrusion detection. In this paper, we discuss the approaches for feature selection, and optimization of parameters of random forests, compare different models, and also discuss the other method for detecting the anomalies across active networks.**

I. INTRODUCTION

Although a wide range of security technologies such as information encryption, access control, and intrusion prevention can be deployed to protect network-based systems, there are always many undetected intrusions. For example, firewalls cannot prevent internal attacks. Thus, Intrusion Detection Systems (IDSs) play a vital role in network security.

*Network Intrusion Detection System*s are built on a network segment to compare captured network data with a set of known malicious signatures, which is stored in a file. In network IDS network cards are used to sniff at all packets of the network segment. A typical network IDS consists of one or more sensors, with a console which aggregates and analyzes sensors' data. *Host-based IDS*s are deployed on a host computer, which continuously monitors processes that run inside the host. The HIDS monitors the changes to a number of variables on the host. The variables can be system processes, Registry entries, CPU usage, file access and integrity checks. Any deviation from the threshold values and changes in file result in the IDS sending an alert. The changes in the key files are verified for evidence by the system integrity verifier. If a match found the HIDS logs and an alert is send to the network or security administrator. *A hybrid IDS* is a combination of a host and a network IDS. The implementation varies from IDSs placed at any critical network aggregation to entry points and on hosts used for conducting business-critical functions.An Intrusion is a planned unauthorized attempt to access information, manipulate information, and/or commit acts such that the system becomes unreliable or unusable. Intrusion detection is performed through two main checks – Anomaly Detection, and Signature Recognition. In Signature Recognition, the IDS checks for a "signature" or pattern. IDSs contain databases with strings based on patterns of known hacker techniques. IDSs examine the network traffic, looking for patterns which resemble any attack pattern. Anomaly Detection is performed by capturing statistical anomalies with reference to a set of activities like CPU utilization, file activity, user logins, disk activity, etc. When there is a deviation from the baseline, the IDS is able to detect it as an intrusion.

II. NETWORK ANOMALY DETECTION WITH UNSUPERVISED OUTLIER DETECTION

Anomaly detection is a very important aspect in Network Intrusion Detection Systems (NIDSs). Most anomaly based NIDSs make use of supervised algorithms [1, 2, 3]. The performances of these algorithms highly depend on attack-free training data. However, it is difficult to obtain such training data in real world network environment. Apart from this, change in network environment or services, results in the change of patterns of normal traffic. This actually results in high false positive rate of supervised NIDSs. Unsupervised outlier detection [4, 5, 6] can overcome the drawbacks of supervised anomaly detection. Therefore, one of the efficient data mining algorithms [7] called random forests algorithm in anomaly based NIDSs is applied here. This algorithm detects outliers in the network traffic dataset without the attack-free training data. The framework of anomaly based network intrusion detection is discussed here. In the framework, random forests algorithm builds patterns of network services over traffic data. Intrusions can be detected by determining outliers related to the built patterns.

III. OVERVIEW OF THE FRAMEWORK

The proposed framework applies random forests algorithm to detect the intrusions. The framework is shown in Figure 1. The NIDS captures the network traffic and constructs dataset by pre-processing. After that, the random forests algorithm is used to build the service-based patterns. With the built patterns, we can find outliers related to each of them. The system will

978-1-4244-1923-4/08/$25.00 ©2008 IEEE

raise alerts if such outliers are detected. Due to the high computational requirements by the outlier detection algorithm, online processing is not suitable in real network environment. So this system involves offline processing once the network traffic is captured.

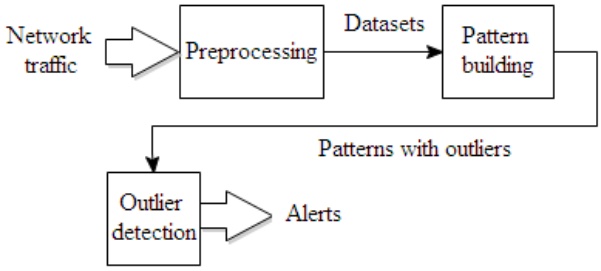

Figure 1 : The framework of the unsupervised anomaly NIDS

Dataset and Preprocessing

The DARPA dataset is commonly to test most of IDSs. The KDD'99 dataset is a subset of the DARPA dataset prepared by Sal Stofo and Wenke Lee [8]. This dataset is a preprocessed dataset consisting of 41 features (e.g., protocol type, service, and flag) extracted from the tcpdump data in the 1998 DARPA dataset. This dataset can be used without further time-consuming preprocessing and different IDSs can be compared with each other by using the same dataset.A complete KDD '99 dataset containing 4,898,431 connections with attacks is used for experimentation.

The attacks in the dataset fall into four categories [9]: DoS (Denial of Service), R2L (unauthorized access from a remote machine), U2R (unauthorized access to root privileges), and probing as shown in Table 1.To generate new datasets for our experiments, the dataset is separated into two pools according to the labels. One includes normal connections while the other includes attacks. Then, the labels are removed from the pools. However, the data labeled by service is required to build patterns of services, so service feature in the dataset is used as label. As a result, all the data contains 40 features and is labeled by service.

Table 1 : Attack categories

0	Normal
1	Probe
2	DoS
3	U2R
4	R2L

Random Forests

The random forests [7] are an ensemble of unpruned classification or regression trees whose literature in relevance to intrusion detection is found in [15, 17]. Random forest generates many classification trees. A tree classification algorithm is used to construct a tree with different bootstrap sample from the original data. When the formation of forest is completed, a new object which is to be classified is taken from each of the tree in the forest. A vote is given by each tree which indicates the decision of the tree decision about the class of the object. The forest selects the class with the most votes for the object.

The main features of random forests algorithm [14] are listed as follows:

1. It is unsurpassable in accuracy among the current data mining algorithms.
2. It shows efficient performance on large data sets with many features.
3. It can give the estimate of what features are important.
4. It has no nominal data problem and does not over-fit.
5. It can handle unbalanced data sets.

In random forests, there is no necessity for cross validation or any test set to get an unbiased estimate of the test error. Since each tree is constructed using the bootstrap sample, approximately one-third of the total cases are omitted out of the bootstrap samples and they do not appear in the training. These are called out of bag (oob) cases and they are used to get a run-time unbiased estimate of the classification error as trees are added to the forest.

Feature selection

With the raw audit data of network traffic it is not possible to detect intrusion. Hence, feature construction is needed to extract a set of features which can detect intrusions effectively. Usually, the construction is based on each connection. Each connectionless packet (*e.g.*, UDP packet) can also be treated as a connection. There are three types of features for network connection records used in NIDSs [10, 13, 16]:*Intrinsic features*: Intrinsic features describe the basic information of connections, such as the duration, service, source and destination host, port, and flag.*Traffic features*: These features are based on statistics, such as the total number of connections to the same node as the current connection within a time window.*Content features*: These features are constructed out of the payload of traffic packets instead of packet headers, such as number of failed logins, whether logged in as root or not, and number of accesses to control files.Feature selection is one of the important steps in building NIDSs. The number of intrinsic features is fixed, since the number mainly depends on the information of packet header. However, traffic features and content features can be constructed using different methods. Though hundreds of traffic and content features can be designed, only some of them are essential for separating intrusions from normal traffic. Unessential features not only increase computational cost, but also increase the false

alarm rate, especially for some algorithms that are sensitive to the feature count. "Deciding upon the right set of features is difficult and time consuming" [11]. Now-a-days, features are designed by domain knowledge experts. Thus, an approach is needed to automate the feature selection. We employ variable importance calculated by random forests. The features that have higher value of variable importance have more effect on classification. Therefore, we choose the features with the higher value of variable importance in the NIDS.

Unsupervised Outlier Detection

We can detect intrusions by finding unusual activities or outliers. There are two types of outliers in the proposed NIDS. The first type is the behaviors that deviate significantly from others when the same network service is in use. The second type is the behaviors whose patterns belong to other services other than their own service. For instance, if an ftp activity is classified as http service, the activity will be indicated as an outlier. Random forests algorithm uses proximities to find outliers whose proximities to all the other cases in the entire dataset are generally small. The proximities are one of the most useful criteria in random forests algorithm [7]. After the forest is constructed, all cases in the dataset are added to each tree in the forest. If cases x and y are in the same leaf of a tree, their proximity is incremented by one. Finally, normalization of the proximities are done by dividing by the total number of the trees.

According to random forests algorithm, outliers are cases whose proximities to all the other cases in the dataset are small [12]. Outlier-ness denotes a degree of being an outlier which calculated over proximities. *class(k)* = *j* indicates that class *k* belongs to class *j*. *prox(n,k)* gives the proximity between cases *n* and *k*. The average proximity from case *n* in class *j* to case *k* (the rest of data in class *j*) is computed as: $\overline{P}(n) = \sum_{class(k)=j} prox^2(n,k)$ N denotes the number of cases in the dataset. The raw outlier-ness of case *n* is defined as: $N / P(n)$.

For each class, the median and the absolute deviation of all raw outlier-ness are determined. The median is subtracted from each raw outlier-ness. The result of the subtraction is divided by the absolute deviation which gives the final outlier-ness. If the outlier-ness of a particular case is large, the proximity is small, so the case is determined to be an outlier. To detect outliers in a set of network traffic, patterns of services are built over the dataset. Then, the proximity and outlier-ness for each activity is calculated. An activity that exceeds a specified threshold will be marked as an outlier.

IV. ANOMALY BASED DETECTOR FOR ACTIVE NETWORKS

We use a novel intrusion detection making use of the features of active network technology without any overhead on additional hardware or topology changes as the solution can be deployed in selective routers of the active network with ease. The system involves a three layer architecture consisting of protocol decoder, misuse detector and anomaly detector which is based on traffic pattern analysis.

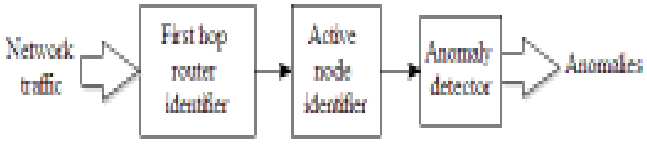

Figure 2 : Anomaly detector for active networks

This module is given more importance to reduce the false alarm rate and improve the detection efficiency. The following sub modulest hop router identifier, active node identifier and anos are designed for implementation namely the firmaly detector.

First hop router generates a message digest (M1) using MD5 on the payload of the packet assuming that the packet might not have been intruded in the first hop itself and stores this value using a hash function generated using the source and destination address to conserve the storage space so that packets are hashed according to the packet flow in the network. It also calculates another digest value using MD5 based on its address so as to detect attack of first hop router address field itself and includes it in the capsule. It is assumed that the message digest functions are global functions known to all active routers for this IDS application.

Checking at each router is unnecessary and hence we find the active routers in the system using the mutual path algorithm. Mutual path algorithm computes all mutually exclusive paths for a flow in the network.Each packet carries a field for first hop router address and some routers in the path are identified as active routers for IDS implementation. These routers on receiving the packets generate M2 value based on first hop field in the capsule and verify the capsule M2 field. On guaranteeing with the first hop router address, the active router computes the payload digest M1 and sends a probe packet directly to the first hop router to confirm the value before forwarding the packet. Else it drops the packet intimating the source as a means of prevention. There is overhead in sending the probe packets. However, as only the active routers send the probe packets to the source, the overhead is reduced. Thus anomaly is detected.

V EXPERIMENTS AND RESULTS

Performance of our system is calculated on the basis of the number of trees constructed during the training phase. More the number of trees constructed more the amount of accuracy with only a small reduction in the performance. Figure 3 shows the comparison of Random Forests algorithms with two different models (5 and 20 trees). As can be seen, with the increase in the number of trees used in the forest, the false positive rate decreases while determining attacks.

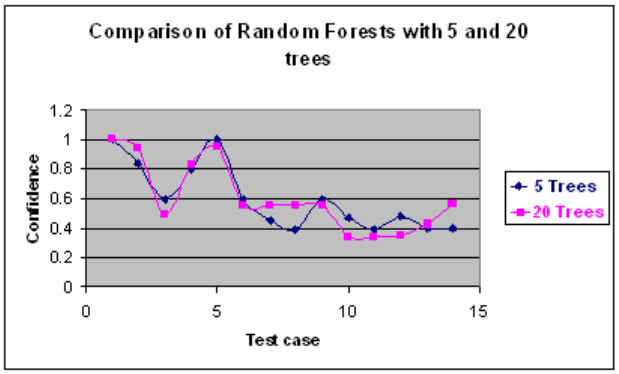

Figure 3 : Confidence values in Random Forests

Figure 4 shows the comparison of Random Forests algorithms with several different models. It shows that the execution times for different models do not vary very significantly. As the number of trees increases, the execution time for a given test set increases. This reduction in performance is negligible upon consideration of reduction in the rate of false positives.

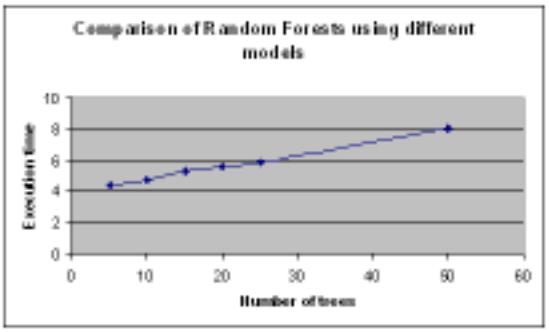

Figure 4 : Execution times in Random Forests

VI.CONCLUSION

In this paper, Random Forests algorithm is employed in NIDSs to improve detection performance. To increase the rate of minority intrusion detection, we build the balanced dataset by over-sampling the minority classes and down-sampling the majority classes. Random forests can build patterns efficiently over the balanced dataset, which is much smaller than the original one. Using Random Forests algorithm, a model is generated using the training set, which is used to classify the test cases. The Anomaly based detector method is used to initially identify the active routers in the system. It is in the active routers that packets are checked for corruption and attacks detected using Random Forests. We plan to implement both misuse and anomaly detection. Misuse detection can reduce false positive rate and anomaly detection can detect novel intrusions. We have started the work on multiple classifier architecture whose overall performance will be higher than the performance of each classifier built by random forests.

REFERENCES

[1] Susan M.Bridges, and Rayford B. Vaughn, "Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection*"*, Proceedings of the National Information Systems Security Conference (NISSC), Baltimore, MD, October, 2000

[2] Q.A.Tran, H.Duan, and X. Li, "One-class Support Vector Machine for Anomaly Network Traffic Detection", The 2nd Network Research Workshop of the 18th APAN, Cairns, Australia, 2004.

[3] Daniel Barbarra, Julia Couto, Sushil Jajodia, Leonard Popyack, and Ningning Wu, "ADAM: Detecting Intrusions by Data Mining", Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security, T1A3 1100 United States Military Academy, West Point, NY, June 2001.

[4] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data", Applications of Data Mining in Computer Security, Kluwer, 2002.

[5] Rasheda Smith, Alan Bivens, Mark Embrechts, Chandrika Palagiri, and Boleslaw Szymanski, "Clustering Approaches for Anomaly Based Intrusion Detection", Walter Lincoln Hawkins Graduate Research Conference 2002 Proceedings, New York, USA, October 2002.

[6] Kingsly Leung and Christopher Leckie, "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters", Australasian Computer Science Conference, Newcastle, NSW, Australia, 2005.

[7] L. Breiman, "Random Forests", Machine Learning 45(1):5–32, 2001.

[8] KDD'99 datasets,The UCI KDD Archive, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Irvine, CA, USA, 1999.

[9] Wenke Lee and Salvatore J. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems", ACM Transactions on Information and System Security (TISSEC*)*, Volume 3, Issue 4, November 2000.

[10] W. Lee and S. J. Stolfo, "Data Mining Approaches for Intrusion Detection", the 7th USENIX Security Symposium, San Antonio, TX, January 1998.

[11] Jiong Zhang and Mohammad Zulkernine, "Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection", IEEE International Conference on Communications, Volume 5, Issue, June 2006, Page(s):2388 – 2393.

[12] J. Zhang and M. Zulkernine, "Network Intrusion Detection Using Random Forests", Proc. of the Third Annual Conference on Privacy, Security and Trust, St. Andrews, New Brunswick, Canada, October 2005.

[13] Snort, Network Intrusion Detection System, http://www.snort.org

[14] Breiman, L. "Consistency for a Simple Model of Random Forests". *Technical Report 670*, Statistics Department, University of California at Berkeley, 2004

[15] Adele Cutler and Leo Breiman, RAFT (RAndom Forest Tool), ttp://www.math.usu.edu/~adele/forests/cc_graphics. htm

[16] Eleazar Eskin, Andrew Arnold, Michael Prerau, eonid Portnoy, and Salvatore Stolfo. "A Geometric Framework for Unsupervised Anomaly Detection: Detecting intrusions in Unlabeled Data." Applications of Data Mining in Computer Security, 2002.

[17] WEKA software, Machine Learning, The University of Waikato, Hamilton, New Zealand. http://www.cs.waikato.ac.nz/ml/weka/