ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

Analyzing Knowledge Based Feature Selection to Detect Remote to Local Attacks

Mya Thidar Myo Win, Kyaw Thet Khaing

***Abstract*— Intrusion Detection (ID) is the most significant component in Network Security System as it is responsible to detect several types of attacks. The IDS commonly deals with a large amount of data traffic, which involves irrelevant and redundant features. The feature selection is one of the prominent factors that influence the quality of IDS. We observe that performing feature selection improves the attack detection accuracy as well as the efficiency of the system. In our experiments, we performed manual feature selection, using our domain knowledge with analyzing the nature of the attack. We compare the results of manual feature selection, automatic feature selection and without feature selection for R2L attack. Feature selection finding a subset of features to improve classification accuracy. These features can be used to uniquely identify a specific attack from all the connections. Experimental result on the KDD cup 99 benchmark network intrusion detection dataset demonstrates that the proposed approach achieved high attack detection accuracy. Random Forest is applied on reduced feature set and classification. It is highly accurate classifier. Our proposed work as good as others and time saving for the classification accuracy for R2L attacks.**

***Index Terms*—feature selection, Intrusion Detection, KDD'99 Dataset, R2L.**

I. INTRODUCTION

The growth of network intrusions on large enterprise networks continues to increase. Thousands of hackers try to probe and attack computer networks each day. These attacks range from relatively benign ping sweeps to sophisticated techniques exploiting security vulnerabilities [1].Intrusion detection is the task of detecting and responding to this kind of computer misuse, by detecting unauthorized access to a computer network [2]. Intrusion detection systems are "systems that collect information from a variety of system and network sources, and then analyze the information for signs of intrusion and misuse". In other words, an IDS is a device, typically a designated computer system, that monitors activity to identify malicious or suspicious alerts. An IDS can be compared with a spam filter, that raises an alarm if specific things occur [3].

Before introducing intrusion detection system as a defense tool, selecting necessary features is important.

Manuscript received May, 2013.

***Mya Thidar Myo Win**, Faculty of Information and Communication Technology. University of Technology Yatanarpon Cyber City, Myanmar.*

***Kyaw Thet Khaing**, Hardware Department, University of Computer Studies Yangon, Myanmar.*

It is because the success of the intrusion detection system depends on the decision upon the set of features that the system is going to use for detecting the attacker especially on detecting the attack. Furthermore, extraneous features inside the network traffic or audit data may be harder for the intrusion detection system to detect the suspicious behavior of attack.

Feature selection improves classification by searching for the subset of features, which best classifies the training data. The features under consideration depend on the type of IDS, for example, a network based IDS will analyze network related information such as packet destination IP address, logged in time of a user, type of protocol, duration of connection etc. Network intrusion detection systems operate at the periphery of the networks and are, thus, overloaded with large amount of network traffic, particularly in high speed networks. [4]. A network intrusion detection system which uses two features 'logged in' and 'number of file creations' to classify network connections as either *normal* or *attack*. When these features are analyzed in isolation they do not provide significant information which can help in detecting attacks. However, analyzing these features together can provide meaningful information for classification. This is because, a particular user may or may not have privileges to create files in the system or the system may detect anomalous activity by calculating deviation in the current profile and then comparing it with the previously saved profile for that particular user.

Network Intrusion Detection System (NIDS) which can analyzes connection level feature such as 'service invoked at the destination' in order to detect attacks. [5] When this feature is analyzed in isolation, it is significant only when an attacker requests for a service that is not available at the destination and the system may then tag the connection as a *Probe* attack. However, if this information is analyzed in combination with other features such as 'protocol type' and 'amount of data transferred between the source and the destination'; the audit data provides significant details which help in improving classification. The relationships between different features in the observed data, if considered by an intrusion detection system during classification can significantly decrease classification error, thereby improving the attack detection accuracy.

The rest of the paper is organized as follows: Section 2 presents an overview of related works. Section 3 gives the features within the KDD data set and Section 4 gives overview selected attacks in intrusion detection field. Section 5 discusses the detection rate of our system when applied to the KDD 99 data.

II. Related Works

Huang, Pei and Goodman [6], where the general problem of GA optimized feature selection and extraction is addressed. In their paper, Huang, et al. applies a GA to optimize the feature weights of a KNN classifier and choose optimal subset of features for a Bayesian classifier and a linear regression classifier. Experiments in their paper show that the performance of all these three classifiers with feature weighing or selection by a GA is better than that of the same classifiers without a GA. They conclude that performance gain is completely dependent on what kind of classifier is used over what type of data set.

Srinivas and Sung [7] presented the use of support vector machine (SVM) to rank these extracted features, but this method needs many iterations and is very time-consuming. In the research of detection model generation, it is desirable that the detection model be explainable and have high detection rate, but the existing methods cannot achieve these two goals.

Chou et al. [8] presented an information theoretic feature selection algorithm on both high and low dimensional feature spaces with correlation analysis; thus verifying the performance of the IDS using a combination of k-nearest neighbor, fuzzy clustering and Damper-Shafer theory. A rough set based parallel genetic algorithm hybrid model is considered to address the important features in building an IDS is considered by Mahmud et al. in [9].

An ensemble approach [10] helps to indirectly combine the synergistic & complementary features of the different learning paradigms without any complex hybridization. The ensemble approach outperforms both SVMs MARs & ANNs. SVMs outperform MARs & ANN in respect of Scalability, training time, running time & prediction accuracy.

This paper [11] focuses on the dimensionality reduction using feature selection. The Rough set support vector machine (RSSVM) approach deploy Johnson's & genetic algorithm of rough set theory to find the reduct sets & sent to SVM to identify any type of new behavior either normal or attack one.

The paper [12] use the feature selection algorithm of random forests, because the algorithm can give estimates of what features are important in the classification.

III. KDD'99 Dataset and Properties

Every record in the KDD 1999 data set presents 41 features which can be used for detecting a variety of attacks such as the *Probe*, *DoS*, *R2L* and *U2R*. Although the KDD'99 datasets provide several attacks and several features not all of the features contribute to an attack. Therefore it is important to study the dataset and select relevance features that contribute to a particular attack. This will make the IDS systems more efficient by reducing the computational cost. Features are grouped into four categories [13]:

- Basic Features: Basic features can be derived from packet headers without inspecting the payload.
- Content Features: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts;
- Time-based Traffic Features: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval;
- Host-based Traffic Features: Utilize a historical window estimated over the number of connections in this case 100 – instead of time. Host based features are therefore designed to assess attacks, which span intervals longer than 2 seconds.

However, using all the 41 features for detecting attacks belonging to all these classes severely affects the performance of the system and also generates superfluous rules, resulting in fitting irregularities in the data which can misguide classification. Hence, we performed feature selection to effectively detect different classes of attacks. We now describe the nature of the selected attack for selecting features why some features were chosen over others.

IV. Description of Selected Attacks and Their Relevant Features

In this section, signatures of the selected attack will be analyzed. The aim will be to extract relevant features from signatures that must be selected to conclusively observe the attack in a networked environment.

A. *Ftp_write Attack*

The Ftp-write attack is a Remote to Local User attack that takes advantage of a common
anonymous ftp misconfiguration. The anonymous ftp root directory and its subdirectories should not be owned by the ftp account or be in the same group as the ftp account. If any of these directories are owned by ftp or are in the same group as the ftp account and are not write protected, an intruder will be able to add files (such as an rhosts file) and eventually gain local access to the system [14,15].

The ftp-write attack is a remote to local user attack that takes advantage of a common anonymous ftp misconfiguration. The ftp directory and its subdirectories should not be owned by the ftp account or be in the same group as the ftp account. If any of these directories are owned by ftp or are in the same group as then ftp account and are not write protected, an intruder will be able to add files (such as a .rhosts file) and eventually gain local access to the system. We could detect this attack easily due to the site-specific policy that no file could be written in ftp directory.

B. *Guesspassword Attack*

This is similar to a dictionary attack, where the attacker makes repeated attempts to login by guessing the password. The behavioral specification had a specification which limited the number of login attempts and flagged an attack, when the number exceeded 3[1]. The guest attack is not amenable to detection using a specification of normal behavior because of the fact that the detection of the attack requires the knowledge that attackers commonly try the user, password pairs of guest/guest and guest/anonymous. The attacks simulated by Lincoln Labs involve only two such attempts, with the second attempt ending in a successful login [15]. We therefore encoded this knowledge about attacker

behavior in our specifications, and were then able to detect all instances of the guest attack.

C. SNMPguess and SNMPget Attack

The Simple Network Management Protocol, SNMP, is a commonly used service that provides network management and monitoring capabilities. SNMP offers the capability to poll networked devices and monitor data such as utilization and errors for various systems on the host. SNMP is also capable changing the configurations on the host, allowing the remote management of the network device. The protocol uses a community string for authentication from the SNMP client to the SNMP agent on the managed device [16]. The SNMP exploit takes advantage of these default community strings to allow an attacker to gain information about a device using the read community string "public", and the attacker can change a systems configuration using the write community string "private".

In the case of the snmpguess attack, the attacker sends infinity of SNMP request with various community name and the victim replies for each one, by sending an empty SNMP message. Each couple of request reply is considered as a SNMP connection independently from the others. To differentiate the SNMP attack traffic from that normal we used the two attributes “num_failed_login” and "logged_in" which belong to the 41 attributes of the transformation function. The first one, “num_failed_login”, count the number of failed login in a session and the second one, "logged_in", indicate that the user of the session in progress presented the good password or not. For the “num_failed_login” parameter that counts the number of times an attacker gives a bad password. The value is incremented when the attacker gives a bad community name this is detected when the victim answers by an empty SNMP message (empty SNMP response event) [17]. So it is SNMPguess.

For the "logged_in" parameter the value is set at when the attacker gives the good community name, i.e,when we detect that the victim answers by a non-empty SNMP message (SNMP response event). After having found the right community name, the attacker will observe the community. SNMP traffic generated by the attacker will then be regarded as pertaining to the same SNMP session where the attacker guessed the good password. SNMP records will have by consequence the same value of the number of failure login attribute. The snmpgetattack traffic is recognized as normal because the attacker logs in as he was a non malicious user since he has guessed the password.

After analyzed the proposed attack, we select 15 features out of the total of 41 features by applying the union operation on the feature sets of the four individual attack classes. The features selected for detecting *R2L* attacks are presented in Table I.

TABLE I. SELECTED FEATURES FOR PROPOSED ATTACKS

Feature Number	Feature Name
1	Duration
2	Protocol
3	Service
4	Flag
5	Src_ bytes
6	dst _bytes
10	hot
11	num failed logins
12	Logged in
13	num compromised
17	num file creations
18	num shells
19	num access files
21	is host login

V. PERFORMANCE COMPARISON WITH PROPOSED FEATURES

We have used an open source machine learning framework WEKA [Waikato Environment for Knowledge Analysis] written at University of Waikato, New Zealand [18].The input data for weka classifiers is represented in .ARFF [Attribute Relation Function Format], consisting of the list of all instances with the values for each instance separated by commas. We perform our experiments with the benchmark KDD 1999 intrusion data set [13]. The raw data from the KDD 99 is first partitioned into four groups (input data set), DoS attack set, Probe attack set, R2L attack set and U2R attack set. For each attack set different connection record feature set are selected as attributes. Classification is performed using Random Forests (RF) algorithm. It is one of the most successful ensemble methods that is fast, robust to noise, and does not over fit. Random forests algorithm is more accurate and efficient on large dataset like network traffic.

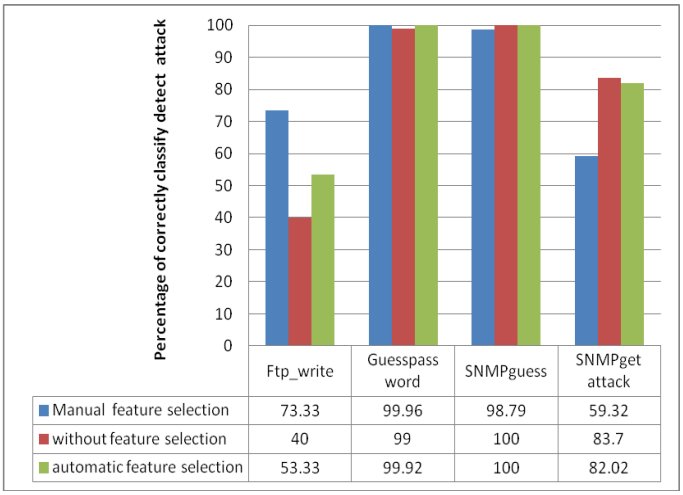

Figure I. Comparison with three experiences of R2L attacks

We perform experiments, both, with and without feature selection and automatic feature selection. Comparison from Figure clearly suggests that a system implementing with feature selection is more efficient and more accurate in detecting attacks. For training our system to detect *R2L* attacks, we select 22544 records from KDD dataset. To test the model, we select all the *R2L* records. In Figure I show the comparison with Classification rate of attack categories in three experiences. We get experiments with manual Feature Selection with the features shown in Figure I. In the second experiment, we give the results for detecting R2L attacks when we use all the 41 features. We use the same data as used in the previous experiment. In automatic feature selection, Weka tool [18] is used for feature reduction. CfsSubsetEval with Best first approach is applied on the training dataset to obtain the important features for the classification process.

In Figure I show that manual feature selection is better performance than without feature selection and automatic feature selection except SNMPguess and SNMPget attack. These attacks are misclassifying with normal while we can see visualized classification error in weka tool. It showed a classification rate of 73.33 % for Ftp_write attack and 99.96%. for Guesspassword attack. However the SNMPget classification rate of 59.32% is low compared to the classification rate of other category of attacks. It should be noted that most of the machine learning algorithms offered an acceptable level of classification rate for DoS and Probe attack categories as they exhibit multiple connections over a short period of time, while demonstrated poor performance for the R2L and U2R categories as these attacks are embedded in their data packets itself and do not form a sequential pattern unlike DoS and Probe attacks. This makes their detection by any classifier a difficult task. Inspite of this, our approach gained good classification rate.

VI. CONCLUSION

In this paper, we compared manual feature selection with automatic feature selection and without feature selection for intrusion detection. First, feature relevance is performed by analyzing the nature of selected attack. It analyses the involvement of each feature to classification and a subset of features are selected as relevant features. Then Random Forest is applied on reduced feature set and classification. As compared to the existing techniques, our proposed work as good as others and time saving for the classification accuracy for R2L attacks. As a future work, we would like to extent the system to real time data capture and online detection of intrusions.

ACKNOWLEDGMENT

I would like to thank my supervisor and all of my teachers for their helpful comments in improving our manuscript. We would like to thank the anonymous reviewers for their thorough reviews, and constructive suggestions which significantly enhance the presentation of the paper.

.

REFERENCES

[1] Jackson, T., Levine, J., Grizzard, J., and Owen, H. (2004). An investigation of a compromised host on a honeynet being used to increase the security of a large enterprise network. In Proceedings of the 2004 IEEE Workshop on Information Assurance and Security. IEEE.

[2] Proctor, P. (2001). The practical Intrusion Detection Handbook. Prentice Hall.

[3] Pfleeger, C. and Pfleeger, S. (2003). Security in computing. Prentice Hall.

[4] S.S.Kim and A.L.N.Reddy, "Statistical techniques for detecting traffic anomalies through packet header data", IEEE/ACM Transaction on Networking, Vol. 16, no. 3, pp.562-575, January 2008.

[5] Eunhye Kim,Seungmin Lee, Kihoon Kwon and Sehun Kim ., "Feature Construction Scheme for Efficient Intrusion Detection System". Journal of Information Science and Engineering 26, 527-547 (2010)

[6] Huang, Z., Pei, M., Goodman, E., Huang, Y., and Li, G.Genetic algorithm optimized feature transformation: a comparison with different classifiers. In Proc. GECCO 2003, pp. 2121-2133.

[7] Srinivas, M., Sung, A., "Feature Ranking and Selection for Intrusion Detection". Proceedings of the International Conference on Information and Knowledge Engineering, 2002.

[8] Chan TS, Yen KK and Luo J., "Network intrusion detection design using feature selection of soft computing paradigms", International journal of computational intelligence ,2008, 4(3):196-208.

[9] Mahmud WM, Agiza HN and Radwan E., " Intrusion detection using rough sets based parallel genetic algorithm hybrid model", In: Proc. of the world congress on Engineering and computer Science (WCECS-2009), USA.

[10] Srinivas Mukkamala, Andrew H. Sung, Ajith Abraham, "Intrusion Detection Using an Ensemble of Intelligent Paradigms",Journal of Network & Computer Applications ,pp-1-15, 2004.

[11] Shilendra Kumar, Shrivastava ,Preeti Jain, "Effective Anomaly Based Intrusion Detection Using Rough Set Theory & Support Vector Machine(0975-8887), Vol:18,No:3, March 2011,DOI: 10.5120/2261-2906.

[12] Jiong Zhang and Mohammad Zulkernine, "Network Intrusion Detection Using Random Forests", School of Computing Queen's University, Kingston Ontario, Canada K7L 3N6

[13] KDD-CUP 1999 Data, http://kdd.ics.uci.edu/ databases /kddcup99 / kddcup99.html.

[14] P. Uppuluri and R. Sekar, "Experiences with Specification-based Intrusion Detection".(2000)

[15] MIT Lincoln Labs, 1998 DARPA Intrusion Detection Evaluation. Available on: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.

[16] The SNMP FAQ www.faqs.org/faqs/by-newsgroup/comp/comp.protocols.snmp.html

[17] Amine Bsila, Sylvain Gombault, Abdelfateh Belghith. "Improving traffic transformation function to detect novel attacks" 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications March 25-29, 2007 – TUNISIA.

[18] Weka tool [online] Available http:// www.cs.waikato.ac.nz/ml/weka.

.