# Detection and Classification of Attacks in Unauthorized Accesses

Mya Thidar Myo Win, and Kyaw Thet Khaing

***Abstract*—** Intrusion Detection System (IDS) is an effective security tool that helps to prevent unauthorized access to network resources by analyzing the network traffic and classifying the records as either normal or anomalous. The features of KDD Cup '99 attack dataset are reduced for each class of attacks performed manual feature selection; using our domain knowledge with analyzing the nature of the attack. And then the reduced feature set, classified with Random Forest classifier, Naive Bayes and k-nearest neighbour. Our selected features improve the attack detection accuracy as well as the efficiency of the system in detect with any classifier. So our selected features are very effective in attack detection. In this paper, classification techniques are used to predict the severity of attacks over the network. I have compared with KDDCUP 99 databases from MIT Lincoln Laboratory.

***Keywords*—**Classifier, feature selection, Intrusion Detection, KDD'99 Dataset.

## I. Introduction

INTRUSION detection is becoming an important technology that monitors network traffic and identifies network intrusions such as anomalous network behaviors, unauthorized network access and malicious attacks to computer systems.An Intrusion detection system is a system for detecting intrusions and reporting them accurately to the proper authority. [1]

The IDSs can also be classified into two categories depending on where they look for intrusions. A host-based IDS monitors activities associated with a particular host, and a network based IDS listens to network traffic. There are two general categories of intrusion detection systems: misuse detection and anomaly detection. Misuse detection depends on the prior representation of specific patterns for intrusions, allowing any matches to them in current activity to be reported.

Patterns corresponding to known attacks are called signatures, also giving rise to the term signature-based detection. These systems are unlike virus-detection systems; they can detect many known attack patterns and even variations; thereof but are likely to miss new attacks. Regular updates with previously unseen attack signatures are necessary [2].

Mya Thidar Myo Win, Faculty of Information and Communication Technology. University of Technology Yatanarpon Cyber City, Myanmar. (e-mail: myathidarmyowin@gamil.com).

Kyaw Thet Khaing, Hardware Department, University of Computer Studies Yangon, Myanmar. (e-mail: kyawthetkhaing.ucsy@gmail.com).

Anomaly detection identifies abnormal behavior. It requires the prior construction of profiles for normal behavior of users, hosts or networks; therefore, historical data are collected over a period of normal operation. IDSs monitor current event data and use a variety of measures to distinguish between abnormal and normal activities. These systems are prone to false alarms, since user's behavior may be inconsistent and threshold levels will remain difficult to fine tune. Maintenance of profiles is also a significant overhead but these systems are potentially able to detect novel attacks without specific knowledge of details. It is essential that normal data used for characterization are free from attacks [2].

Network-based IDS monitors traffic by capturing and analyzing network packets. Advantages of network-based IDSs are: (i) the deployment of these systems has little impact on the existing network; (ii) little effect on the normal network operation and are relatively easy to upgrade, and (iii) robust in the face of attacks and can be made invisible to attackers. On the other hand, the disadvantages are: (i) during peak-traffic periods some packets may go unprocessed and attacks undetected; (ii) encrypted information cannot be analyzed; (iii) attack attempts may be detected but hosts must usually then be investigated manually to determine whether or not they were penetrated and damage caused, and (iv) attacks involving fragmentation of packets can cause these IDS to crash [3].

Host-based IDS monitors network traffic of a particular host and some system events on the host itself. One may be installed on each host or simply on some chosen critical ones within a network. Advantages of host-based IDSs are: (i) some local events on hosts can only be detected; (ii) raw data are available for analysis in non-encrypted form, and (iii) software integrity checks can be used in the detection of certain types of attack (e.g. Trojan horse). In addition, it has the following disadvantages: (i) more complex to manage; (ii) may be disabled if host is attacked and compromised; (iii) not suitable for network attacks involving distributed scans and probes; (iv) can be disabled by overload attacks (e.g. denial of service); (v) for large amounts of information to be processed, local storage may be necessary, and (vi) use host's own computing resources at a cost to performance [3].

Before introducing intrusion detection system as a defense tool, selecting necessary features is important. 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. After understanding the relation and influence of the feature, we propose a set of minimum feature that can be used in

detecting the selected attack. Typical and relevant features must be observed present in the KDD data set that can help with the detection of the selected attacks. The attacks can be classified with Random Forest, Naïve Bayes and k -nearest neighbor for the robustness of our selected features.

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

Jayshri R.Patel [4] presented performance of four selected decision tree classification algorithms for ranked intrusion detection data is evaluated and investigated. From the experiment & result analysis it is very clear that the performance of Random Forest is better as it correctly identifies more number of instances than other.

Younes Chihab[5] present a comparative study between five data mining algorithms to come up finally with the proposition of a hybrid classifier based on Random Forest and Naïve Bayes algorithm. This method provides an effective distinction between different types of intrusions which allows us to customize the treatment given to each type of intrusion. These methods are tested using the KDD'99 database.

According to studies presented in the literature of the field [6-9], it can be said that the detection rate for different attack types is higher by using different feature sets for each attack type category instead of using the same features for all the attack types. In addition it can be said that by using less features it is possible to reach higher detection rate than by using all of the available 41 features in KDD cup dataset.

Reema Patel[10] presented compares various data mining techniques used to implement an intrusion detection system such as Decision Trees, Artificial Neural Network, Naïve Bayes, Support Vector Machine and K- Nearest Neighbour Algorithm by highlighting advantages and disadvantages of each of the techniques. Finally, a discussion of the future technologies and methodologies which promise to enhance the ability of computer systems to detect intrusion is provided and current research challenges are pointed out in the field of intrusion detection system.

Panda and Patra [11] have compared the performance of Naïve Bayes with the Neural Network approach and found its suitability in building an intrusion detection model.

Xianfei Zhang [21] proposed a system analysis of k -nearest neighbor and its improved algorithm, and presents the k -nearest neighbor algorithm based on fuzzy integral. The new method uses fuzzy integral to fuse the k-nearest neighbor of training samples, which avoids independency demand of D-S theory and improves performance of text classification.

## III. KDD'99 Dataset and Properties

KDD Cup '99 intrusion detection datasets [12] which are based on DARPA '98 dataset provides labelled data for researcher working in the field of intrusion detection and is the only labelled dataset publicly available. The details of KDD dataset are given in the subsequent section. The KDD dataset is generated using a simulation of a military network consisting of three target machines running various operating systems and traffic. Finally, there is a sniffer that records all network traffic using the Tcpdump format. The total simulated period is seven weeks. Normal connections are created to profile that expected in a military network and attacks fall into one of the four categories:

- Denial of Service (Dos): Attacker tries to prevent legitimate users from using a service.
- Remote to Local (R2L): Attacker does not have an account on the victim machine, hence tries to gain access.
- User to Root (U2R): Attacker has local access to the victim machine and tries to gain super user privileges.
- Probe*:* Attacker tries to gain information about the target host.

There are 41 features for each connection, which are detailed in Table I. Specifically, "a connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows from a source IP address to a target IP address under some well-defined protocol". Features are grouped into four categories:

- Basic Features*:* Basic features can be derived from packet headers without inspecting the payload.
- Content Features: Domain knowledge is used to access the payload of the original TCP packets. This includes features such as 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 instead of time. Host-based features are designed to access attacks, which span intervals longer than 2 seconds.

TABLE I
KDD DATASET FEATURE (SUMMARIZED FROM [12])

| No | Features Name | No. | Features Name |
|---|---|---|---|
| 1 | duration | 22. | is guest login |
| 2 | protocol | 23. | count |
| 3 | service | 24. | srv_ count |
| 4 | flag | 25. | serror rate |
| 5 | source bytes | 26. | srv serror rate |
| 6 | Destination bytes | 27. | rerror rate |
| 7 | land | 28. | srv rerror rate |
| 8 | wrong | 29. | same srv rate |
| 9 | urgent | 30 | diff srv rate |
| 10 | hot | 31. | srv diff host rate |
| 11 | failed logins | 32 | dst_host _count |
| 12 | logged in | 33. | dst_ host _srv_count |
| 13 | # compromised | 34. | dst_host_same_srv_ rate |
| 14 | root shell | 35. | dst_ host_ diff_srv _rate |
| 15 | su attempted | 36. | dst_host _same src_port rate |
| 16 | # root | 37. | dst host srv diff host rate |
| 17 | file creations | 38. | dst host serror rate |
| 18 | # shells | 39. | dst host srv serror rate |
| 19 | # access files | 40. | dst host rerror rate |
| 20 | # outbound cmds | 41. | dst host srv rerror rate |

| | | | |
|---|---|---|---|
| 21 | is hot login | | |

## IV. Description Of Attacks In Unauthorized Accesses And Their Relevant Features

In this section, we describe the nature of the attacks for selecting features why some features were chosen over others. The aim will be to extract relevant features from signatures that must be selected to conclusively observe the attack in a networked environment.

User to root attack involves unauthorized access to local super user privileges by a local unprivileged user. *U2R* attacks are the attacker pretends as a legitimate user of the system without authorization and then exploits the system's vulnerabilities to get root access of that system. For example, the attacker may exploit a system's vulnerabilities to gain root privileges and install a backdoor program onto a system for future access. The result may cause the system crash or make the system execute the attacker's program as if it is part of the system's original programs. U2R attacks may result in significant loss of time and money for many organizations. Therefore, the detection of such attacks is very important to ensure security in computer and network systems [13].

### A. Buffer _Overflow

Buffer overflow exploits the vulnerability of a system that does not correctly perform a boundary check of user's input data before copying it to a fixed length memory buffer. Once the vulnerability is found, the attacker can supply excess data into the insufficiently sized memory buffers and therefore possibly corrupt the data and thus make the service crash. Furthermore, the attacker can add executable data into the stream and remotely activate it to gain unauthorized access when the buffer overflows. Example can be seen such as installing a backdoor program on the vulnerable system for future use [14].

### B. Loadmodule

Loadmodule attack is window system server to load two dynamically loadable kernel drivers into the currently running system and to create special devices in the directory to use those modules. Because of a bug in the way the loadmodule program sanitizes its environment, unauthorized users can gain root access on the local machine. This attack can be identified either by performing bottleneck verification with a host based intrusion detection system, or by keyword spotting with a network based intrusion detection system [15].

### C. Perl

The Perl attack is a User to Root attack that exploits a bug in some Perl implementations Suidperl is a version of Perl that supports saved set-user-ID and set-group-ID scripts. In early versions of suidperl the interpreter does not properly relinquish its root privileges when changing its effective user and group IDs. On a system that has the suidperl, or sperl, program installed and supports saved set-user-ID and saved set-group-ID, anyone with access to an account on the system can gain root access [15].

### D. Rootkit

Rootkits pose a serious and growing threat to computer systems today. "Rootkit" was originally used to refer to a toolkit developed by the attacker, which would help conceal his presence on the compromised system. The rootkit is typically installed after the attacker has obtained root level privileges on the compromised system [16]. Rootkits are installed on a system that has already been compromised via some other method. The most popular method of gaining entry into the system is by exploiting software vulnerabilities, which usually happens without the user's knowledge or interaction.

Most commonly exploited software vulnerabilities exist in the operating system and browser software. Users often lag in updating their systems with released security patches, which plays out to the attacker's advantage. Rootkits might also be injected when the user intentionally downloads software from untrusted web sites. Such software is often bundled with unwanted components, several of which might be malicious.

After analyzed the proposed attack, we select 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 attacks are presented in Table II.

TABLE II
SELECTED FEATURES FOR ATTACKS IN AUTHORIZED ACCESSES

| No | Attack | Selected Feature |
|---|---|---|
| 1 | Buffer_Overflow | 1,2,3,4,5,6,13,14,16,17,18,19,21 |
| 2 | Loadmodule | 1,2,3,4,5,6,10,17 |
| 3 | Perl | 1,2,3,4,5,6,14 ,17,18 |
| 4 | Rootkit | 1,2,3,4,5,6,10,12,13,14,17 |

## V. Experimental Result

Kddcup99 dataset produced by Lincoln Laboratory at MIT where each record has been specified as normal or attacked one with specific type of attacks. We have used an open source machine learning framework WEKA [Waikato Environment for Knowledge Analysis] written at University of Waikato, New Zealand [17].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 [12]. 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.

Using all the 41 features and only using selected features of each attack. We compare the experimental results in the following three classifiers: Random Forest, Naïve Bayes and k-nearest neighbour based on the order of intrusion detection steps for evaluation. A random forest is an ensemble of decision trees which will output a prediction value. That operates by constructing a multitude of decision trees at training time and outputting. Random forests as defined by Leo Breiman [18]. Is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the

forest. Each decision tree is constructed by using a random subset of the training data.

One of the strengths of Naive Bayes technique is that it requires a small amount of information during the learning phase. It is based on Bayes' theorem [19], are widely used in the classification domains. The operating principle of this algorithm is based on the assumption that each class of examples is independent [20].

K-Nearest Neighbour (k-NN) is instance based learning for classifying objects based on closest training examples in the feature space. It is a type of lazy learning where the function is only approximated locally and all computation s deferred until classification. The k-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms [10].

In the experiments, we randomly selected 9711 normal connections and each attack connections form the training set. For test set, we randomly selected 9711different normal connections and different attacks connections show in table.

TABLE III
DATA USED FOR TRAINING AND TESTING IN THE EXPERIMENTS

| | Training | Testing |
|---|---|---|
| Normal | 9711 | 9711 |
| Buffer_Overflow | 30 | 20 |
| Loadmodule | 9 | 2 |
| Perl | 3 | 2 |
| Rootkit | 10 | 13 |

In Figures show that the detection of attack in unauthorized accesses using random forest, Naïve Bayes and k-nearest neighbor. The result compare the percentage of correctly classify instances with selected features and all features of each attack. The correctly classification rate of buffer overflow with selected feature is higher than classification with all features in three kind of classifiers. In figure 1, while 95% of buffer overflow attack is correctly classify with selected feature using random forest, there are no correctly classify with all features. In figure 2 and 3, detection of buffer overflow with selected features is better performance than detected with all features. Moreover Loadmodule and Perl are full percentage with our selected features in three kind of classifier. However, rootkit is more correctly classify with all features in Naive Bayes and KNN because the rootkit attack traffic is recognized as normal. However, rootkit is more correctly classify with all features in Naive Bayes and KNN because the rootkit attack traffic is recognized as normal. Detection with selected features is better performance than detection with all features in any three classifiers. It is showed that our selected features are robustness in any classifier.

These detection results only using the selected attributes almost remain the same or even become better than those using all the 41 features .This shows that many of the 41 attributes are irrelevant and only a smaller set of attributes is required to extract from raw network traffic for detection of individual attacks. Our selected features based on attack nature are very affected in attack detection and experience shows that high accuracy and time saving.

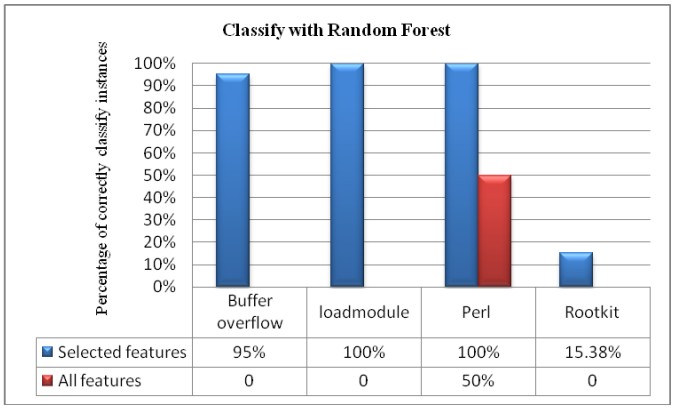


Fig. 1 Correctly classify instances of attacks using Random Forest

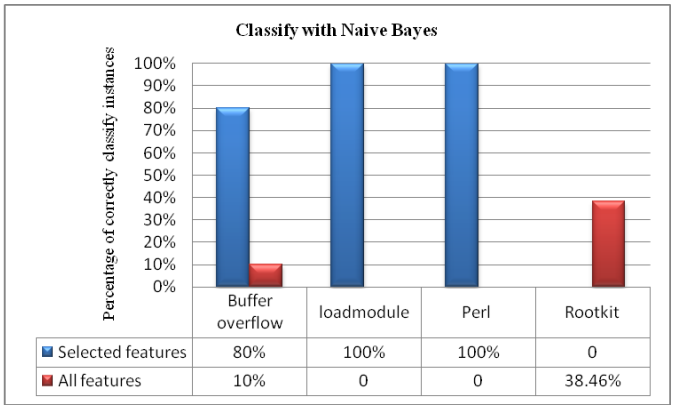


Fig. 2 Correctly classify instances of attacks using Naïve Bayes

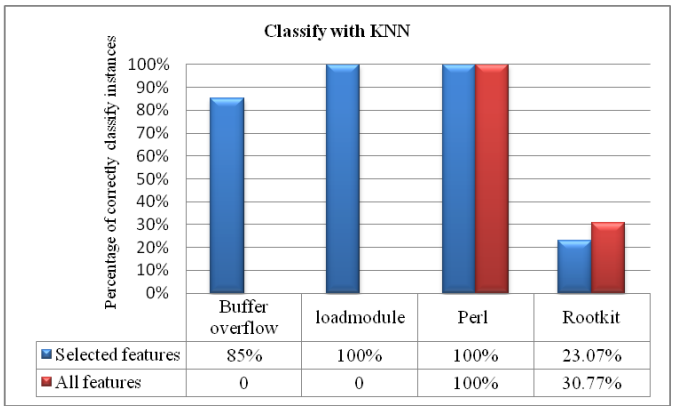


Fig. 3 Correctly classify instances of attacks using K-Nearest Neighbour

## VI. CONCLUSION

In this paper, we compared the result of detection of attacks with selected features and all features. 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, Naïve Bayes and k-nearest neighbor are applied on classification. Our proposed work as good as others and time saving for the classification accuracy for 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] J. P. Anderson, Computer Security Threat Monitoring and Surveillance, Technical Report, James P. Anderson Co., Fort Washington, PA, April 1980.

[2] Verwoed T. and Hunt R., "intrusion detection techniques and approaches," Elsevier: computer communications, Vol.25, No.10, pp: 1356-1365, 2002.

[3] Chobrolu S., Abraham A., Johnson P., "feature deduction and ensemble design of intrusion detection systems," Elsevier computers & security, Vol.24, pp: 195-307, 2005.

[4] Jayshri R.Patel, "Performance Evaluation of Decision Tree Classifiers for Ranked Features of Intrusion Detection", Journal of Information,Knowledge and Research in Information Technology, ISSN: 0975 – 6698, VOLUME – 02, ISSUE – 02.

[5] Younes Chihab, Abdelah Ait Ouhman, Mohammed Erritali, Bouabid El Ouahidi , "Detection & Classification of Internet Intrusion Based on the Combination of Random Forest and Naïve Bayes", Younes Chihab et.al / International Journal of Engineering and Technology (IJET), ISSN : 0975-4024 Vol 5 No 3 Jun-Jul 2013.

[6] P. Gifty Jeya, M. Ravichandran, C. S. Ravichandran, "Efficient Classifier for R2L and U2R Attacks", International Journal of Computer Applications (0975 – 8887)Volume 45– No.21, May 2012

[7] A. A. Olusola, A. S. Oladele and D. O. Abosede, "Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features". Proceedings of the World Congress on Engineering and Computer Science, vol. 1, Oct-2010.

[8] Kayacik, H.G., Zincir-Heywood, A.N. and Heywood, M.L. (2006). Selecting Features for Intrusion Detection: A Feature Analysis on KDD 99 Intrusion Detection Datasets.

[9] Dr.S.Siva Sathya, Dr. R.Geetha Ramani and K.Sivaselvi, "Discriminant Analysis based Feature Selection in KDD Intrusion Dataset", International Journal of Computer Applications (0975 – 8887), Volume 31– No.11, October 2011.

[10] Reema Patel, Amit Thakkar, Amit Ganatra, "A Survey and Comparative Analysis of Data Mining Techniques for Network Intrusion Detection Systems", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012.

[11] Panda, Mrutyunjaya and Patra, Manas Ranjan , " Network Intrusion Detection using Naïve Bayes", International journal of computer science and network security, Dec'30-2007, pp.258-263.

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

[13] Iftikhar Ahmad, Azween B Abdullah, Applying Neural Network to U2R Attacks, 2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2010), October 3-6, 2010, Penang, Malaysia.

[14] Craig Sheppard, "Buffer Overflows and Application Security",February 3, 2004, SANS Institute 2004,

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

[16] M. Mahoney and P. Chan, "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection", Proceeding of Recent Advances in Intrusion Detection (RAID)-2003, Pittsburgh, USA, September 2003.

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

[18] L. Breiman. Random Forests. Machine Learning, 45(1):5-32, 2001. http://dx.doi.org/10.1023/A:1010933404324

[19] Nakache, D. (2007). Extraction automatique des diagnostics à partir des comptes rendus médicaux textuels. Laboratoire CEDRIC - équipe ISID. Paris, Conservatoire National des Arts et Métiers: 219.

[20] Han, J., M. Kamber Concepts d'exploration de données et des techniques [M] X. Meng, Pékin: Appuyez sur l'industrie mécanique, 2005 196 201.

[21] Xianfei Zhang, Bicheng Li, Xianzhu Sun, "A k-Nearest Neighbor Text Classification algorithm Based on Fuzzy Integral", Sixth International Conference on Natural Computation (ICNC 2010)