

# Intrusion Detection Using Datamining Techniques

Anshu Veda(04329022)  
KReSIT,IIT Bombay

Prajakta Kalekar(04329008)  
KReSIT,IIT Bombay

Anirudha Bodhankar(04329003)  
KReSIT,IIT Bombay

## I. OBJECTIVES

To come up with an Anomaly-Detection based Intrusion Detection System. The goal was to try out various data mining approaches and analyze the results of the same when used for Anomaly Detection. Schemes such as Outlier Detection for *Network based IDS* and *Prediction of system calls* for Host based IDS, were tried out.

## II. APPROACH

We tried to detect anomalies using both host-based and network-based approaches. We used Anomaly Detection for both Host-based as well as Network-based IDS approaches.

### A. Network-Based

1) *Input data preprocessing*: The network-based approach relies on the tcpdump data as input, which gives per packet information. We used data a 1.5GB dataset that we collected over a 24hrs span. We used as the normal data. This data was preprocessed as suggested in [1], grouping records corresponding to one connection. Following this, Content-based, Time-based and Connection-based features were extracted from the data.[1] The attack data was obtained by simulating a TCP-SYN flood [5] using packETH. The attack data was also pre-processed using the same procedure as that used for the normal data.

2) *Datamining approach used*: Next, we explored a number of techniques like Association Rule Mining and Frequent Episode rules[2]. Association Rule mining usually is very slow and though once a popular technique, its being replaced by other powerful techniques like clustering and classification. Then we came across a recent paper [1] which advocated the use of outlier detection technique for detecting the anomalous data points in datasets. Clustering was the first choice because the dataset was *huge* and *multidimensional*. We used the K-Means algorithm for this. The idea was to train a K-Means clusterer using *Normal* datasets and cluster the normal behavior points. For the test data set, the probability of its belonging to the most probable cluster, was computed. If this was *below* a threshold, the instance was flagged as anomalous. This approach did not give us very good results. As a consequence, even the data points corresponding to attack data were being assigned to clusters with a very high probability. We were unable to configure K-Means Algorithm implemented in Weka in such a way that the distance between the centroid of a cluster and data-point be used to calculate its probability of belonging to that cluster. On analysis we concluded that perhaps the attack that we used for simulation(the SYN Flood attack) was affecting only a few fields of the data-points -

like num\_bytes, num\_syn and the like. We tried to cluster on some of these specific attributes, and it increased accuracy too. However, this implied the use of *Domain Knowledge* of the fields that would be affected, which is not the aim of anomaly detection techniques.

Next, we tried out the KNN technique for Outlier Detection. For each test data-point, the distance from its Nearest Neighbor was computed. If this distance is found to be greater than some prespecified threshold, the point is marked as an Outlier. The next step was to establish this threshold. We obtained the threshold by finding out the distance of each training instance from its Nearest Neighbor (or average of distance from k-Nearest Neighbors). The maximum of these distances was estimated as the value for the threshold. We modified Weka's IBK algorithm for computing the distance from the nearest neighbor. This approach gave us good results, and the results obtained have been discussed below.

### B. Host-Based

1) *Input*: Normal traces of system calls were generated using the *Stace* utility for the *sshd* process. This data was preprocessed so as to generate normal sequences of system calls by sliding a window of size 4 over the input file. An attack was launched over the system, by attempting to connect to a system using ssh and by trying out different passwords for the same.

The attack data was split into different time windows (of 0.5 hours each). The detection algorithm runs over one such window at a time and detects the presence of anomalous regions.

2) *Datamining approach used*: The technique we adopted for anomaly detection was *Prediction of the  $i^{th}$  system call* for a record containing a sequence of  $n$  system calls. The predicted value was compared with the actual value. If the value was found to be different, then the confidence of prediction of the value is taken into consideration. All these confidence scores are added up to compute the total *misclassification score*. If this misclassification score crosses a threshold, then the region is classified as an anomalous region. We used Classification Technique for prediction since the data had few dimensions, equal to the size of the sliding window. The different options we considered for Classification were Decision Trees, SVM, Naive Bayes and meta-learners formed by the combination of these techniques. Out of these, Decision Trees gave us the best results. However, this maybe due to the lack of tuning of the other Classification Models such as SVM.

### III. RELATED WORK

We adopted the technique for data processing for NIDS as proposed in [1]. We also adopted the NIDS technique of Outlier Detection using K-Means and KNN as proposed by [1]. The Prediction of  $i^{th}$  System Call technique has been explained in great detail in [2]. We followed the same, except that instead of using Rule Mining for Prediction, we used Classification techniques. Apart from the above mentioned techniques - we also surveyed techniques such as Mining of Traffic episodes [3] and Frequent Episodes in Event Spaces [4]

### IV. SUMMARY OF RESULTS

The following results were observed:

#### A. Network based IDS

1) *K-Means*: The Outlier threshold for probability of belonging to a cluster was selected as 0.7. For the dataset we tested on, this threshold gave the best results. As indicated by Table I, the method suffered from a large number of false negatives, when all the features were used for the Clustering. As indicated by Table II, number of false negatives

	P	N
P	0.2	0.8
N	0.1	0.9

TABLE I  
CONFUSION MATRIX USING K-MEANS (CONSIDERING ALL ATTRIBUTES)

dropped when only a few of the relevant attributes namely num\_bytes and num\_syn were used for Clustering. However as mentioned earlier, domain knowledge cannot be exploited in Anomaly Detection. For this technique, the Outlier threshold for probability of belonging to a cluster was selected as 0.8. For the dataset we tested on, this threshold gave the best results.

	P	N
P	0.55	0.45
N	0.05	0.95

TABLE II  
CONFUSION MATRIX USING K-MEANS (CONSIDERING ONLY RELEVANT ATTRIBUTES)

2) *KNN*: The results obtained using KNN varied for the different values of threshold that were selected by the system. The presence of false negatives indicated that **the the selection criteria of the threshold value needs to be refined**. It indicates that in certain cases, the maximum of the nearest neighbor distances in the training set maybe less than the nearest neighbor distance of some anomalous test instances. This was one aspect of interest to us. We tried variations of the method for selecting the threshold value like averaging of

the distances of K-Nearest values, taking average of values of not the top k, but top k except the nearest, except the second nearest etc. The results varied for different test sets. The best results we obtained were for a value of threshold equal to 0.196. As indicated by Table III, the results obtained using KNN were better as compared with those obtained using K-Means.

	P	N
P	0.8	0.2
N	0.0	1

TABLE III  
CONFUSION MATRIX USING KNN FOR NETWORK BASED SYSTEMS

#### B. Host based IDS

We tried out different values of  $i$ , the index of the system call in the record of system calls, to be predicted. The results obtained by varying these were nearly the same. As recommended in [1], we selected the  $\frac{n^{th}}{2}$  system call for prediction. For prediction, the method that we selected was Decision Trees, as it gave a better performance as compared with SVM, Naive Bayes and other meta-learners. But we believe that adequate training of the other Classification models could yield a better performance. Another factor influencing anomaly detection was the selection of the misclassification score. By varying this value, we realized that a threshold of approximately 0.2 gave a good performance. The best result we obtained was using Decision Trees. As Table IV indicates, the number of True positives is considerably high.

	P	N
P	0.87	0.13
N	0.05	0.95

TABLE IV  
CONFUSION MATRIX USING DECISION TREES FOR HOST BASED SYSTEMS

### V. EFFORT/TIME DISTRIBUTION

The major areas in which time was invested:

- *Data Preprocessing*  
Anshu : Survey of NIDS preprocessing  
Prajakta : Survey of the HIDS preprocessing  
Anirudha : Writing of preprocessing scripts for tcpdump and Stace logs  
Time spent per person : 1 day
- *Attack Simulations*  
Anshu : Study of Network-based Attack simulations  
Prajakta : Study of Host-based attack simulations  
Anirudha : Simulation of attacks using tools and scripts  
Time spent per person : 6-7hrs
- *Implementation*  
Anshu : K-Means method, and a part of Host based technique

Prajakta : Modification of Weka's IBk for KNN, and the implementation thereof, and a part of Host based technique.

Anirudha : Writing of testing scripts

Time spent per person : 8-10hrs

- Apart from the above, some time was spent in gaining domain knowledge and surveying the various data mining strategies[6],[7],[1] that could be used.

## VI. STATUS OF PROJECT

### A. Completed part

We have been able to detect attacks using both Network and Host-based IDS, which is the hybrid technique that is recommended for a perfect IDS. Our IDS has been tested with real data (IITB logs).

### B. Things to do

- Refining the technique of coming up with a good threshold for the KNN approach IV-A.2
- Tuning of the various Classifiers for Prediction of System Calls technique IV-B
- Testing with bigger datasets.

### C. Challenges faced

The problem of use of Datamining approaches for Intrusion Detection is a much-researched one. Studying the pros and cons of the various techniques and shortlisting the ones to implement was a long and tedious process. We feel there is a lot more for us to study and explore in this area.

## REFERENCES

- [1] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, V. Kumar *A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection*
- [2] W. Lee, S. J. Stolfo *Data Mining Approaches for Intrusion Detection*
- [3] Min Qin and Kai Hwang *Anomaly Intrusion Detection by Internet Datamining of Traffic Episodes*
- [4] H. Mannilla, H. Toivonen, A. I. Verkamo *Discovery of Frequent Episodes in Event Sequences*
- [5] Thesis by Kristopher Kendall. *A database of computer attacks for the evaluation of Intrusion Detection System*
- [6] W. Lee, S. Stolfo, K. Mok. *Mining in a data-flow Environment: Experience in Network Intrusion Detection*
- [7] MinQin&Kai Gwang *Anomaly Intrusion detection by Internet datamining of traffic episodes*