Network Intrusion Detection Evading System using Frequent Pattern Matching

N. B. Dhurpate[#1], L.M.R.J. Lobo[*2]

[#1]M.E. Student & Computer Science and Engineering Department & Solapur University

[#1]Walchand Institute of Technology, Solapur, India

[*2]Head Of Dept & Information Technology Department & Solapur University

[*2]Walchand Institute of Technology, Solapur, India

***Abstract*— Signature based NIDS are efficient at detecting attacks for what they are prepared for. This makes an intruder to focus on the new evasion technique to remain undetected. Emergence of new evasion technique may cause NIDS to fail. Unfortunately, most of these techniques are based on network protocols ambiguities, so NIDS designers must take them into account when updating their tools. This paper presents a framework for evading network intrusion detection system and detection over NIDS using frequent element pattern matching. The core of the framework is to model the NIDS using Adaboost algorithm that allows the understanding of how the NIDS classifies network data. We look for some way of evading the NIDS detection by changing some of the fields of the packets. We use publicly available dataset (KDD-99) for showing the proof of our concept. For real time evasion detection NIDS is build with Apriori algorithm to analyze NIDS robustness with high detection rate accuracy.**

***Keywords*—Intrusion Detection, Evasion, Network security, Apriori algorithm, frequent item set, Adaboost algorithm, NIDS.**

I. INTRODUCTION

Information security underpins the commercial viability and profitability of enterprises of all sizes and the effectiveness. Due to advances in technology, communication and the decentralized nature, it is increasingly difficult to ensure that this information is provided in such a way that its integrity is ensured.

Intrusion Detection Systems (IDS) are software or hardware tools that automatically scan and monitor events that take place in a computer or a network, looking for evidence of intrusion [1]. Network Intrusion Detection Systems (NIDS) just analyze network traffic captured on the network segment where they are installed. NIDS may seek for either anomalous activity (anomaly based NIDS) or known hostile patterns (signature based NIDS) on the network.

For every attack which are known signature is stored in NIDS. Signature based NIDS are efficient for detecting attacks for which they are prepared. If signature is not present the NIDS fail to work properly. These signatures can be easily available. So instead of finding new attacks techniques, attacker focuses on evasion over these signatures. The concept of evasion was first proposed by Ptacek and Newsham [2]. The authors highlighted some ambiguities in network protocols (concretely TCP and IP) that can lead into a situation where NIDS and endpoint systems process packets in a different way. An evasion succeeds if the processing of the packets generates a different representation of the raw data in the NIDS and in the end systems. For the evasion we created a framework.

The aim of our framework is to look for new evasive techniques by analyzing NIDS behavior. We created NIDs by Adaboost and Apriori algorithm. By Adaboost accuracy of classification of traffic into normal and attack packet is shown. For real time intrusion detection and evasion we create NIDS by apriori algorithm. By frequent item set rules are created and those are given to snort for detection of attacks. NIDS is able to detect the attacks for which it is prepared. We are successful in showing evasion over NIDS by changing some fields of attack.

II. STATE OF ART

Evasions on NIDS were first proposed by Ptacek and Newsham in 1998 [2]. In this paper, the authors highlighted the existence of some ambiguities in the TCP and IP protocols, which allow different systems to implement them in a different way. An evasion succeeds when NIDS ignore packets which are going to be processed on the endpoints or vice versa. For example, if the ICMP packet contains some bad checksum packets or malicious field, that protocol does not have an idea what to do with those packets. ICMP protocols either ignore or accept or reject those packets. As shown in Figure 1, an evasion could successful if the NIDS implementation of the ICMP protocol differs from the endpoint system implementation [3].

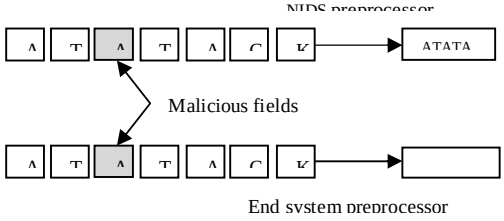


Figure 1: Evasion example

In this example, NIDS and end system treat malicious or bad checksum field differently, as the result NIDS accept the packet while end system reject it. Many techniques have been designed to prevent evasions. Most of them are based on network traffic modification, to remove the ambiguities and establish a common understanding of the protocols for NIDS and endpoints. Our goal is to first model the NIDS then perform the evasion. AdaBoost is the algorithm is used here for constructing a "strong" classifier as linear combination.

III. WORKFLOW

In this frame work, in first half NIDS is created to classify the network data. Here for this Adaboost algorithm is used for classification. And some fields are changed of those packets to check its accuracy. C4.5 [7] is used first and output of that is give to Adaboost for better classification. In second half, for real time evasion detection and modeling NIDS Apriori algorithm is used. Apriori algorithm is based on frequent itemset. Different methods are found out for evasion.

A. Generate the Small Dataset

Adaboost algorithm based NIDS at issue level requires a dataset. This data contain normal (simple web requests, remote connections, web navigation, etc) and intrusive (malicious) traffic. This dataset can be generated by own or we can use the existing datasets. For generating dataset controlled environment is required. Obtained traffic should be exposed to the NIDS, which analyzes the dataset looking for intrusive actions. In Adaboost algorithm first labeling of dataset is done. Attack packets are labeled by +1 and normal are by -1 at the end. Thus, the obtained dataset is composed of registers with the form:

T1,T2,T3,...,TN,L,O

Where each Ti is the field i of the trace (for example, the source port, the flag bits, the amount of data exchanged, etc.), L is the label which indicates the nature of data (normal or attack) and O is the output given by the NIDS (normal or intrusion). The overall dataset is then divided into smaller sets, one being the training subset and the remainder the testing subsets.

B. Model the NIDS

As we know, in our framework Adaboost and Apriori algorithms are used to model the behavior of NIDS. First, values for some parameters are established. This process can be made manually or automatically. This technique consists of performing the Adaboost modeling phase several times, by using different combination of parameters. Each training phase is performed with one fold, using the remainder to test the evolved model. The principal advantage of using this technique is that we explore several combinations of parameter values so we can assure that we are using an optimum values for them, as the training phase is performed with all the different subsets (folds) of the entire dataset, so it does not depends on an initial selection, but in the complete dataset. Once the parameters are fixed, we obtain the NIDS models by training them with the entire training subset. Then, we perform the test of the obtained models using the testing set. Results must be stored to be processed afterwards. Because the Adaboost search is heuristic, it is appropriate to perform the training phase several times, using different random seeds, taking the results for the best individual (the one that has produced the best test results) and the average of the individuals. Using different random seeds covers a bigger searching space. A manual optimization of the model is then performed. The tree model obtained has normally redundant branches or nodes, so performing a pruning phase could be interesting to improve the efficiency of the model.

For modeling Apriori [8] based NIDS we take different session of traffic as input to NIDS. According to support and confidence value rules are generated for frequent item set. These rules based, large item set is then given to snort as input. Snort[4] is an open source NIDS. By this a real time evasion is shown. Snort uses a rule as a signature and it is able to find the attack for what it is prepared. After the attack an alert message is generated by snort.

C. Analysis and Design of Evasive Techniques

After the gaining model, internal structure of NIDS is analyzed for conceiving an idea of its behavior. Mainly, the Model indicates which are the fields that the NIDS takes into account to classify traces. This information is used to perform a brute force modification of those fields. The idea is to automate the process by changing the value of the fields that are present in the model, generating new modified traces. Before changing the value, it should be assured that traces with the new value remain being attacks and still coherent with the protocols [6]. For this purpose, a set of rules must be established and fulfilled, indicating which variables can be changed and which values can be set to them.

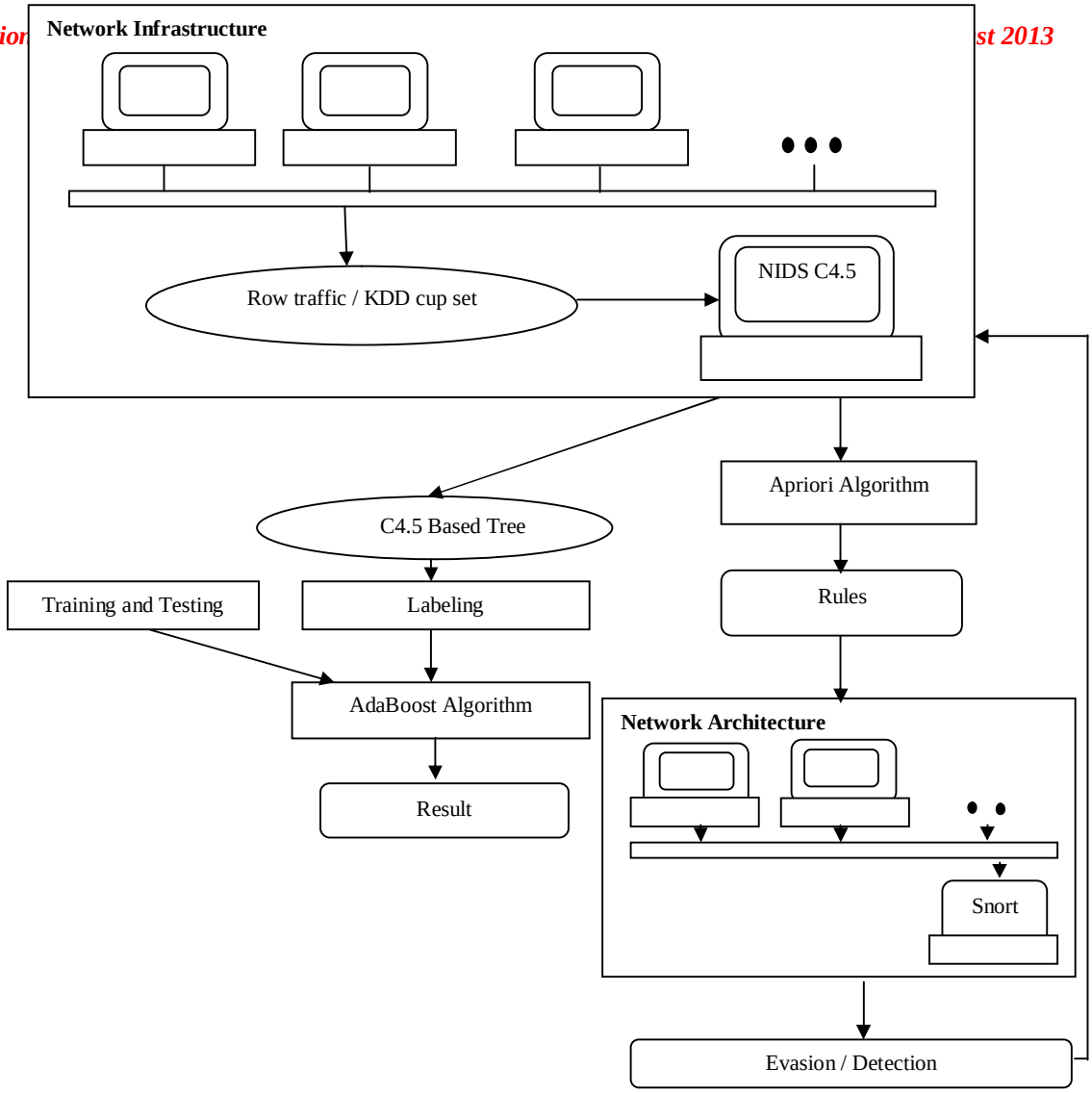


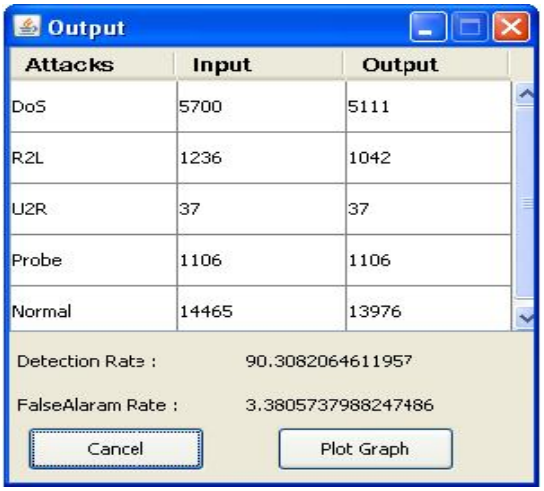
Figure 2: Architecture of NIDS System

New valid values are given for those fields in hostile traces which were previously detected by the NIDS (true positives), establishing a new dataset composed by old and new (modified) traces. Then, the NIDS is applied to those new modified traces New false negatives would indicate that the evasions performed have been successful The process is repeated for each field that appears in the model, and also multiple simultaneous changes (to more than one field at the same time) can be done.

The main objective is to find evasions over the NIDS analyzing the corresponding model. For that purpose, we have created a basic NIDS based on the C4.5 algorithm. This algorithm is a supervised learning classifier whose output is a tree. We use input for NIDS as publicly available dataset KDD-99 which is derived from raw traffic captured during MIT/LL 1998 evaluation. Result of this we get detection rate of our NIDS as 90.30%. Adaboost algorithm classifies attacks into attacks (DOS, R2L, U2R, and probe) and normal packets. Accuracy of system is found out by input and output count.
Detection Rate= (output count/input count) *100
False alarm rate = 1-(output count/input count)

Table 1: Performance of the self-built, c4.5 based NIDS.

IV. PROOF OF CONCEPT SPECIFIC GOALS



We look for evasions by modifying the value of one or more fields of the traces and exposing them to the original NIDS. We must choose fields and values in such a way that the traces remain coherent with protocols. An evasion is considered successful if, after the modification of the trace, the NIDS does not detect it as an intrusion.

For showing an evasion on real time Apriori based NIDS some fields are changed from the attack, so attack remain unnoticed. Here rules are stored in snort, according to that signature intruder do some changes in fields so snort is not able to detect it. Evasion is successive if NIDS fail to give an alert message.

V. EXPERIMENTAL WORK

Figure 2 shows architecture of NIDS. In first step KDD-99 dataset which contain attack and normal traffic is given to C4.5 algorithm through weka [5]. C4.5 in output generate tree. Tree is generated by some attribute value. At each node attributes are given by which tree is further classified. At the leaf node the actual attack is given. On each branch some weight is assigned according to classification attribute. In second step this tree and dataset is given to Adaboost algorithm. Adaboost algorithm has 4 phases labeling, data mining, training, testing. In labeling the normal packet are given -1 value and attack packet +1 at the end. Through data mining some features are extracted. Training phase is performed by taking different fields combination by changing folds. Then the created NIDS is tested for its accuracy. Adaboost algorithm classifies the traffic into 4 types of attacks DOS, U2R, R2L, probe and normal packet. Detection rate and false alarm rate is found out.

For real time evasion NIDS is created using the Apriori algorithm. Different sessions of attacks are given as input to Apriori algorithm. According to support and confidence value rule are generated by apriori algorithm. These rules are given to snort which is open source NIDS. When attack is generated for which signature is stored in snort, it generate alarm. After that we show evasion over NIDS by changing some fields of it. If NIDS failed to generate alarm means evasion is successful. So we found out different types of evasion. The aim is not to break NIDS but try to different evasion techniques.

For this work, we also generate our own dataset. This dataset is generated by capturing traffic at different time of day. According to the attributes dataset is created and saved in .arff format which can be useful for weka input.

VI. CONCLUSIONS

To prevent systems from new attacks, NIDS should be quickly updated. However attacker instead of finding new types of attack tries to remain unnoticed by evading system by using signature. In this work, we create NIDS for classifying traffic in different types of attacks. We have tested our framework by using a simple NIDS based on the C4.5 algorithm over the publicly available datasets. For what real time intrusion detection and evasion NIDS is created by Apriori algorithm. The aim of evasion is not to break the NIDS system but to understand and learn different ways of evasion of system and make system sturdier.

ACKNOWLEDGMENT

It is a pleasure for me to present this paper where guidance plays an invaluable key and provides concrete platform for completion of the paper. It is my great pleasure and privilege to express my sincere gratitude to my guide Prof.L.M.R.J.Lobo for his kind supervision, valuable suggestions, timely guidance, constant encouragement and moral support during the completion of this work.

.

REFERENCES

[1] R. Bace and P. Mell, "NIST Special Publication on Intrusion Detection Systems", 800-31, 2001

[2] T. H. Ptacek and T. N. Newsham, "Insertion, evasion and denial of service: Eluding network intrusion detection," Technical report, 1998.

[3] S. Pastrana, A. Orfila, A. Ribagorda, “A Functional Framework to Evade Network IDS”,IEEE xplore, System Sciences (HICSS), 2011 44th Hawaii International Conference.

[4] S. Peddabachigaria, A. Abraham, “Modeling intrusion detection system using hybrid intelligent systems”, Journal of Network and Computer Applications.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P.Reutemann, I. H. Witten, "The WEKA Data Mining Software: An Update", in SIGKDD Explorations, Volume 11, Issue 1,2009.

[6] Pallavi Dhade, T.J.Parvat, “To Evade Deep Packet Inspection in NIDS Using Frequent Element Pattern Matching”, IJEIT, Volume 2, Issue 1, July 2012

[7] J. Ross Quinlan, “C4.5 Programs for Machine Learning”, Morgan Kaufmann Publishers, Inc., 1993.

[8] Ferenc Bodon, "A fast APRIORI implementation", Informatics Laboratory, Computer and Automation Research Institute.

Ms. Neelam B Dhurpate received B.E degree in Information Technology in 2009 from PUNE University, Maharashtra, India and pursuing the M. E. degree in Computer Science and Engineering in Walchand Institute of Technology, Solapur, India. She is doing her dissertation work under the guidance of Mr. Lobo L.M.R.J, Associate Professor & Head, Department of IT, Walchand Institute of Technology, Solapur, Maharashtra, India

Mr. Lobo l.M.R.J received the B.E degree in Computer Engineering in 1989 from Shivaji University, Kolhapur, India and M. Tech degree in Computer and Information Technology in 1997 from IIT, Kharagpur, India. He is registered for Ph.D in Computer Science and Engineering at SGGS, Nanded of Sant Ramanand Teerth Marathawada University, Nanded, India. Under the guidance of Dr. R.S. Bichkar. He is presently working as an Associate Professor & Head, Department of IT Walchand Institute of Technology, Solapur, Maharashtra, India. His research interests include Evolutionary Computation, Genetic Algorithms and Data Mining.