ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

To Evade Deep Packet Inspection in NIDS Using Frequent Element Pattern Matching

Pallavi Dhade, T.J.Parvat

Abstract*— A set of rules are used Signature based Network Intrusion Detection Systems (NIDS) to detect hostile traffic in network segments or packets , which are so important in detecting malicious and anomalous behavior like known attacks that hackers look for new techniques to go unseen. Most of the techniques involves, in the manipulations of anonymities of network protocol. Now, the detection techniques are developed against most of these elusive and equivocal techniques by means of identifying and recognizing. The presence of new elusive forms may possibly effect NIDS to be ineffective. This paper presents an innovative functional framework to perform modeling and detection over NIDS using frequent element pattern matching. Main, NIDS demonstrated precisely through Apriori algorithm. At this point, the paper consists of looking for avoidances on models are simpler and easier than directly trying to understand the behavior of NIDS. We present a proof of concept showing how to perform deep packet inspection in NIDS using two publicly available datasets. This framework can be used for analyzing, Modeling and detecting the commercial NIDS after elusion.

Index Terms—**Apriori Algorithm, Deep Packet Inspection, Network Intrusion Detection Systems, Frequent Elements Matching, High Speed Network**

I. INTRODUCTION

Network Intrusion Detection Systems (NIDS) just analyze network traffic captured on the network segment. NIDS may look for either anomalous activity (anomaly they are installed. NIDS may seek for either anomalous activity (anomaly based NIDS) or known hostile patterns (signature based NIDS) on the network. Firewalls they do not normally block packets, but aware about the intrusion alarm. This situation focuses on elusions over the signatures of these systems. There are some problems in network protocols that create state where endpoint systems process the packets generates a different demonstration of data in the NIDS and in the end system. If the representation of the data in the NIDS and in the end systems are different then the evasion is successful. Sometimes it is not possible to detect the attacks. Thus, a possible appearance of new elusive techniques would be critical for systems that are supposed to be secure. This is the inspiration and motivation of the work, in which a new approach to watch for elusions over NIDS, giving a proof of concept showing how to perform deep packet inspection in NIDS using two publicly available datasets. This framework can be used for analyzing and Modeling and detecting the malicious behavior in the commercial NIDS. Computer security is defined as the protection of computing systems against threats to confidentiality, integrity, and availability [2]. The goal of Confidentiality (or secrecy) is that information is disclosed only according to policy, integrity means that information is not destroyed or corrupted and that the system performs correctly, availability means that system services are available when they are needed. Computing system refers to computers, computer networks, and the information they handle. Security threats come from different sources such as natural forces, accidents, failure of services (such as power) and people known as intruders. The categories of intruders are the external intruders who are unauthorized users of the system they attack, and internal intruders, who have permission to access the system with some restrictions. The traditional prevention techniques such as user authentication, data encryption, avoiding programming errors and firewalls are used as the first line of defense for computer security. To manage the huge amount of personal, public and critical data we always choose the Information Technology systems, which become a critical component in organizations Identifying such a anomalous behavior and hostile actions, for protecting those systems, which is one of the most important goal in security. Intrusion Detection Systems (IDS) are Software or hardware tools that automatically examine, check and observe events that take place in a computer or a network, looking for indication of intrusion [1].

II. STATE OF ART

There are some uncertainties and doubts in network protocol, which allow different systems to implement them in a different way. An evasion is successful when DPI is not giving attention to packets which are going to be processed on the endpoints system and vice versa. For example, if the ICMP packet contains some inaccurate 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 [1].

In this example, i.e fig.1 the DPI preprocessor accepts the packet containing a malicious field, while the endpoint does not, so the final structure after the preprocessing phase will be different. Many techniques have been designed to prevent evasions. Most of them are based on network traffic

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

modification, to remove the ambiguities and establish a common understanding of the protocols for DPI 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,

$$ht(x) = \sum_{t=1}^{T} \alpha t\, ht(x)$$

of "simple" "weak" classifiers ht(x).

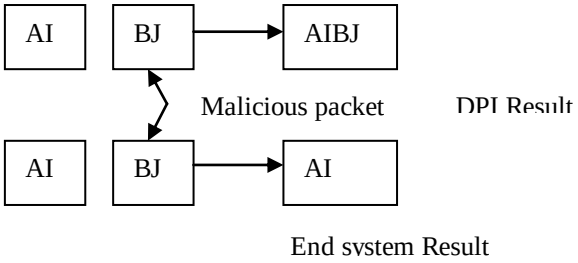


Fig. 1: Elusion example

III. WORKFLOW

Here in this frame work, our goal is satisfied in the second half of the Figure 2 which shows a graphical description of overall framework. The main objective is to look for new evasion techniques on a given NIDS. After elusion, main goal to detect those changes. We use weka tool to obtain a model that classifies as similar as possible to the NIDS. Due to the use of a simple syntax, the Adaboost algorithm has a simpler semantics. Looking for evasive techniques over the model is easier than over the NIDS. If evasions succeed over the model, and given that this model may have a quite similar behavior than then original NIDS, it is likely that the evasions will also succeed over the NIDS. But now the work of our system starts that to detect those elusion. Our framework is composed of a set of tasks described in the following sections.

A. Generate the Small Dataset

The Adaboost modeling process at issue requires a labeled dataset. This dataset must represent as well as possible real traffic. Due to the necessity of generating different traffic profiles, a controlled environment is required. Generated traffic should include normal (simple web requests, remote connections, web navigation, etc) and intrusive (malicious) traffic. Traffic is processed by means of data mining techniques to extract the most significant features. It also needs to be labeled in order to identify it as normal or hostile. Obtained traffic should be exposed to the NIDS, which analyzes the dataset looking for intrusive actions. Output given by the NIDS is appended to its corresponding processed frame [2]. Thus, the obtained dataset is composed of registers with the form:

S1, S2, S3... SN, L, O

Where each Si 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

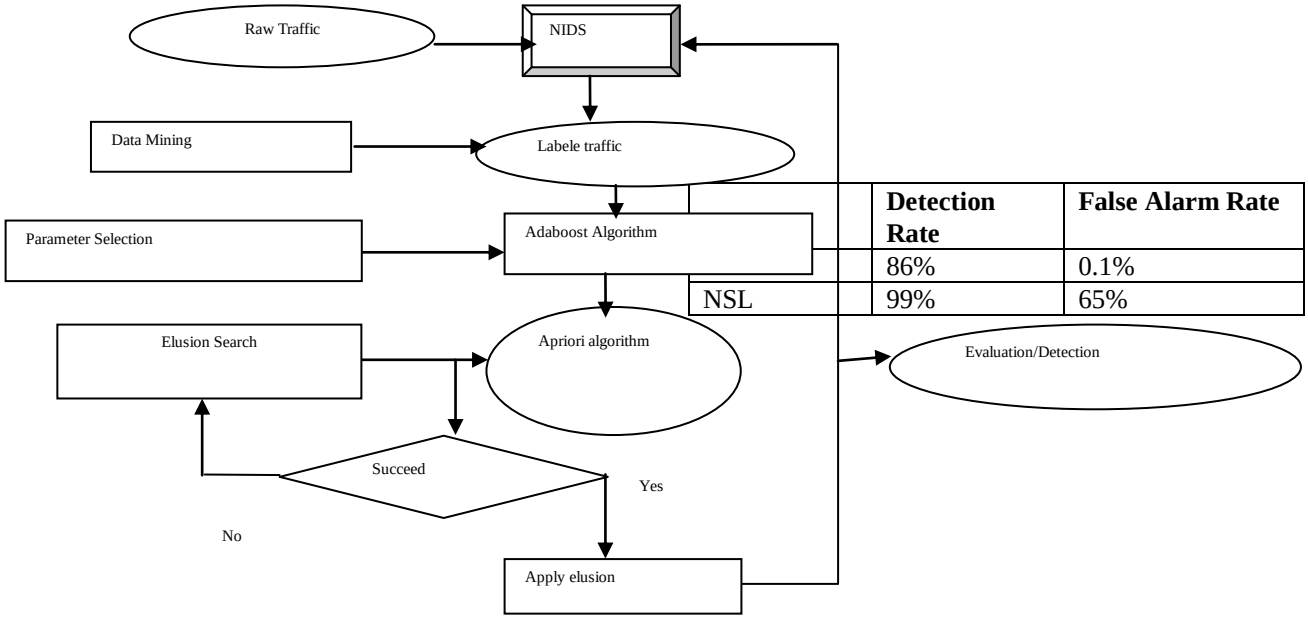
As we know, in our framework Adaboost and Apriori algorithms are used to model the behavior of NIDS. First, values for some parameters are established. This process can be made manually or automatically. This technique consists of performing the Adaboost modeling phase several times, by using different 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 (which has to be considerably bigger than the remainder, used for testing). 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.

C. Analysis and Design of Evasive Techniques

Once the model is obtained, it is analyzed in order to discover some points of the internal structure of the NIDS, thus 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. For that purpose, a set of rules must be established and fulfilled, indicating which variables can be changed and which values can be set to them. 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. Before changing the value, it should be assured that traces with the new value remain being attacks and still coherent with the protocols. For that purpose, a set of rules must be established and fulfilled, indicating which variables can be changed and which values can be set to them. New valid values are given for those fields in hostile traces which were

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

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.



	Detection Rate	False Alarm Rate
	86%	0.1%
NSL	99%	65%

Fig. 2: Functional Framework

D. Specific Proof Concept

The two main objectives of the proof of concept presented are first, 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 [2,4]. This algorithm is a supervised learning classifier whose output is a tree. A simplification of the framework has been made to fulfill our goals. Instead of creating a specific dataset, in this work we have opted to use the only two publicly available datasets that are labeled (as normal or intrusive). Results showed that with an accuracy of 96%, the behavior of a self-built NIDS can be modeled by reducing its complexity. In this work, we improve the study by using a different dataset, the KDD-99 derived from raw traffic captured during MIT/LL 1998 evaluation. The use of an extra dataset corroborates that the accuracy of using Apriori algorithm to model a NIDS is not limited to one scenario and attack, but also to another that uses several kind of attacks. It is obvious that these datasets are both quite old, taking into account the fast growth of the complexity in the Information Technologies. However, they have been widely used in the literature and they provide a huge set of labeled traces. Thus, it is useful for providing insight into the problem at issue and to analyze if the idea is sound. After obtaining models for each dataset, we are challenged to find real evasions over the original C4.5 based NIDS. 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, being still attacks (for example, if we change the bit of some TCP flag in a port scanning attack, we are not evading the NIDS and attacking the endpoint, but transforming the malicious trace into a normal one). For that purpose, need is to analyze the nature of the attacks we are working with. It is also needed that traces to be modified are true positives. An evasion is considered successful if, after the modification of the trace, the NIDS does not detect it as an intrusion. And then the main aim of our system is to perform detection over elusion or change.

IV. EXPERIMENTAL WORK

Figure 3 shows a scheme of the modeling phase. At first the datasets are prepared. The KDD provides both normal and Port scanning traffic, captured in various days at different hours. We use five different raw traffic files, processing them in order to take just TCP traffic. Thus, we establish five datasets containing labeled traces from both malicious and normal nature. These traces are composed of the fields (Ri) of the TCP header. In the case of the KDD dataset, we have taken 10% of the original traces, preprocessed them in order to make the output binary (i.e.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

normal or intrusion) and normalizing the non-numerical fields. We use the weka tool [8] to obtain the C4.5 based NIDS (step 1 in Figure 3). For that purpose, we randomly choose a subset of each dataset to perform the training phase, testing over the remainder. This testing phase provides, for each trace, the output given by the NIDS, i.e. if it has properly classified the trace or not. This information is appended to each trace, obtaining the final dataset (step 2 in the Figure 3). We perform another division of the dataset, in this case to obtain two new different subsets, one to be used in the training phase and another one to test the individuals (step 3 in the Figure 3). Table 1 shows the performance of the NIDS created for both the NSL dataset and the KDD. As can be observed, in the case of the NSL, NIDS are tending to classify the traces as intrusive, so its detection rate and its false alarm rate are both very high. However, the NIDS built with the KDD has lower rates, which indicates that it is more likely to classify the traces as normal. So, given that the NIDS which are going to be modeled are very different in nature, the first goal of our proof of concept, which was to prove the feasibility of using Apriori algorithm to model NIDS goes a step further.

The models are created by first evolving them using a training subset (step 4 in Figure 3), using the remaining subsets to test whether the obtained models have a good performance with different traffic from the one used to evolve them (step 5 in Figure 3). Critical component in C4.5 is that it performs a heuristic search. Accordingly seven different seeds have been used over each training subset, thus obtaining seven different evolved individuals. Then, a Testing process is performed with each individual. In the following section the best and average result for each model is shown. Each individual represents one different NIDS model, and because they must be as simple as possible, a maximum depth of 4 is established.

The models are created by first evolving them using a training subset (step 4 in Figure 3), using the remaining subsets to test whether the obtained models have a good performance with different traffic from the one used to evolve them (step 5 in Figure 3). Critical component in C4.5 is that it performs a heuristic search. Accordingly seven different seeds have been used over each training subset, thus obtaining seven different evolved individuals. Then, a Testing process is performed with each individual. In the following section the best and average result for each model is shown. Each individual represents one different NIDS model, and because they must be as simple as possible, a maximum depth of 4 is established.

Table I. Performance of Self Built Ids Using C4.5

	Detection Rate	False Alarm Rate
KDD	86%	0.1%
NSL	99%	65%

As it was previously stated, one of the goals of this proof of concept is to corroborate that this is a good paradigm to be used when modeling the NIDS. In order to compare with some other techniques, we have obtained models using two different techniques. Concretely, we have used the Naïve Bayes approach, which is a specific Bayesian classifier which assumes strong independence among fields [7] and whose output is not a tree, but a probabilistic model. The second method used is the C4.5 algorithm, which is the one used to create the NIDS under study, but limiting its maximal tree depth to 4. It is obvious that the C4.5 algorithm will reach better results if its maximal depth would not be limited to 4, because it is the algorithm used to obtain the original NIDS. However, this limitation of the maximal depth is needed to assure that the complexity of the models to be compared is similar. In order to evade these models, we are interested in changing traces corresponding to true positives. We analyze the models manually looking for any field that, when changed, will make the NIDS to fail in the detection. It is possible that there is no possible change that causes the evasion of the NIDS. In this case, we should repeat modeling process (by changing some field or the fitness function) in order to obtain another model over which we would look for new evasive method. Now here the evasion is done and the existing NIDS fail to detect the malicious behavior. So again by identifying the fields and the parameter where the changes has already been done, we can perform the better detection with this system. As by generating rules by using Apriori algorithm where by providing support and confidence we identify the parameters which are responsible for evasion. Then the values of those parameters changed and again the detection has been performed which detects the attack which are ignored by the original NIDS after evasion. In this way we improve the detection rate and accuracy.

V.CONCLUSION

Currently, NIDS are prepared to detect a huge variety of attacks. Some of them, like Snort, take into account the possibility of being evaded with the techniques. However, they are not prepared to new evasive forms that can appear. In this paper we present a new framework to look for evasions over a given NIDS. The core of the framework is to model the NIDS using Adaboost Algorithm obtain an easier to understand individual which works as similar as possible to the NIDS. This model allows the understanding of how the NIDS classifies network data. Once this model is obtained, we can look for some way of evading the NIDS detection by changing some of the fields of the packets. The final aim of using our framework is not to break the detection of the NIDS, but to analyze NIDS robustness with high detection rate accuracy.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

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. I would also like to express my sincere thanks to my internal guide Prof. Mr. T. J. Parvat. Department of Computer Engineering, for his unfaltering encouragement and constant scrutiny without which I wouldn't have looked deeper into my work and realized both our shortcomings and our feats. This work would not have been possible without him.

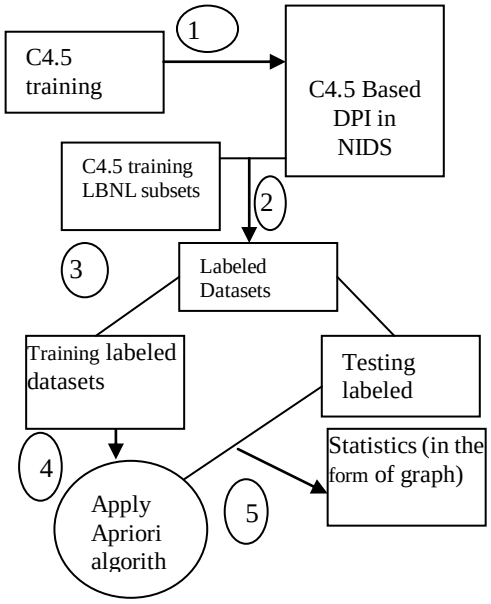


Fig 3. Detailed Designed For Experimental Work

REFERENCES

[1] Xu Kefu, Guo Li, Tan Jianlong, Liu Ping," Traffic aware frequent element matching algorithm for Deep Packet Inspetion",International Conference on Network Security, wireless communication & Trusted Computing, 2010 .

[2] Sergio Pastrana Agustin Orfila Arturo Ribagorda,"A functional framework to evade NIDS", Hawaii International conference on System Sciences, 2011.

[3] J. R. Koza, "Genetic Programming: On the Programming of Computers", International conference on security sciences, 2010.

[4] S. Pastrana, A. Orfila, and A. Ribagorda, "Modeling NIDS evasion with Genetic Programming", on the Proceedings of The 2010 International Conference on Security and Management, SAM 2010.

[5] L. Juan, C. Kreibich, C.-H. Lin, and V. Paxson, "A Tool for Offline and Live Testing of Evasion Resilience in Network Intrusion Detection Systems,",5th international conference on Detection of Intrusions and Malware, and Vulnerability,2008.

[6] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P.Reutemann, H. Witten, "The WEKA Data Mining Software: An Update", An extensive empirical study of feature selection metrics for text classification, 2009.

[7] Po-Ching Lin; Ying-Dar Lin; Tsern-Huei Lee; Yuan-Cheng Lai; "Using String Matching for Deep Packet Inspection," Computer , vol.41,no.4,pp.23-28,April2008.

[8] Kun Huang; Dafang Zhang, "A Byte-Filtered String Matching Algorithm for Fast Deep Packet Inspection,". The 9th International Conference for Computer science 2008.

AUTHOR'S PROFILE

Pallavi Dhade ,Assistant Professor, M.E.(Computer Engineering pursuing),, Department of Computer Engineering, Sinhgad Institute of Technology, Lonavala, Pune, Maharashtra state, India, research area Network security

Prof. T.J.Parvat,,Professor, M.E, P.hd(ng pursuing),, Department of Computer Engineering, Sinhgad Institute of Technology, Lonavala, Pune, Maharashtra state, India ,research area network security