IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

STUDIES OF TECHNIQUE USES A COMBINATION OF STATISTICAL AND FUZZY LOGIC METHODOLOGIES TO ASSESS THE PERFORMANCE OF A NETWORK FEATURE AGAINST A PARTICULAR TYPE OF ATTACK

NABONARAYAN JHA[1], AKHILESH KUMAR[2], & M. KHAN[3]

[1] Department of Mathematics, Patan Multiple Campus, Patan Dhokha, Lalitppur, T. U. Kathmandu & Research Scholar, Department of Mathematics, B R A Bihar University, Muzaffarppur, Bihar, India
[2]Research Scholar, Department of Mathematics, J. P. University, Chhapra, Bihar, India
[3]Department of Mathematics, K. R. College, Gopalganj, J. P. University, Chhapra, Bihar, India

ABSTRACT

To ensure that the selected features reasonably cover all the types of data that can be extracted at the network level, we propose a *Feature Classification Schema* for network features that is also intended to impose the first centralized standard in the research community regarding the various types and names of the network features. In this paper we study various tuning parameters that play a significant role in the performance of each feature. It is widely known that a poor tuning step will lead to poor detection performance, as opposed to a reasonable tuning that will significantly boost the attack detection chances. We explain our proposed feature evaluation technique. The technique uses a combination of statistical and fuzzy logic methodologies to assess the performance of a feature against a particular type of attack.

1. INTRODUCTION

In the case of a wired network those interconnections are realized through cables, routers, and switches, to name a few. In the case of a wireless network, the interconnections are achieved by using towers and antennas. The entities in a network are referred as *hosts*. A host can refer to almost any kind of computer that resides within the network (e.g., server, mainframe, desktop PC, or terminal). The data exchanged between different hosts inside a network is wrapped in *packets*. A packet is the fundamental unit of information carriage in the network. Furthermore, let us consider in general a *connection* as a bidirectional information exchange between two hosts for fulfilling a goal. The proposed feature classification schema is defined for the Transport, Network, and Network Access layers of the TCP/IP Architecture Model. In particular, a host is uniquely identified by its IP address, while a connection (i.e., TCP, UDP, and ICMP1) is uniquely identified by the combination of 6 fields: source IP, destination IP, source port, destination port, protocol, and type of service. However, the schema can be easily extended to the other remaining protocols and layers of the TCP/IP architecture. Let *srcIP*, *dstIP*, *srcPort*, and *dstPort* represent the source IP, destination IP, source port, and destination port of a packet, respectively. Similarly, let *c-srcIP*, *c-dstIP*, *c-srcPort*, and *c-dstPort* represent the source IP, destination IP, source port, and destination port of a connection, respectively. The proposed schema is further used to extract and classify the prominent characteristics (also referred to as `features') of the above *packet*, *host*, *connection*, and *network* abstractions. Figure 1 depicts an overall view of the proposed feature classification schema. Due to the large number of categories, the current work uses acronyms. The naming convention can be followed as a path in the tree depicted by the figure, where each edge carries a letter that is added to the end of the acronym. One of the main dilemmas in intrusion detection is whether or not a packet belongs to a malicious event. If a definite answer can be deducted from the available pool of features, the detection problem is solved. Our classification schema is defined with respect to the currently sniffed packet.

2. FEATURE TUNINGS

Each individual feature, regardless of the category that it belongs to, has multiple tuning variables that dramatically influence its detection performance.

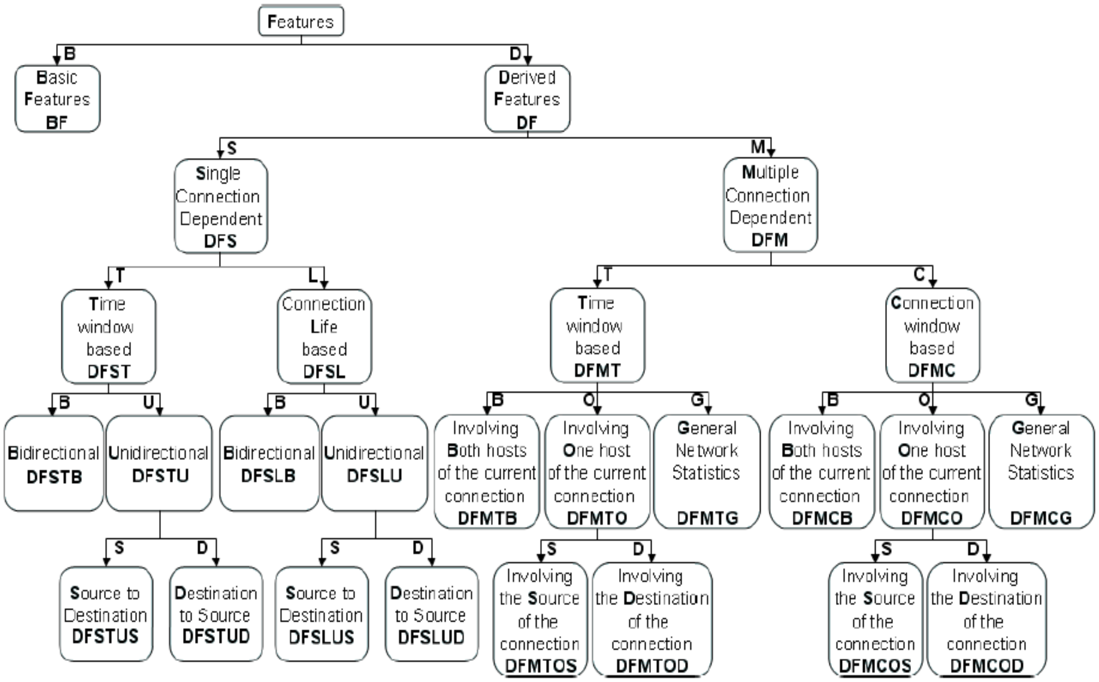
Features
B
Basic Features BF
D
Derived Features DF
S
Single Connection Dependent DFS
M
Multiple Connection Dependent DFM
T
Time window based DFST
L
Connection Life based DFSL
T
Time window based DFMT
C
Connection window based DFMC
B
U
B
U
B
O
G
B
O
G
Bidirectional DFSTB
Unidirectional DFSTU
Bidirectional DFSLB
Unidirectional DFSLU
Involving Both hosts of the current connection DFMTB
Involving One host of the current connection DFMTO
General Network Statistics DFMTG
Involving Both hosts of the current connection DFMCB
Involving One host of the current connection DFMCO
General Network Statistics DFMCG
S
D
S
D
S
D
S
D
Source to Destination DFSTUS
Destination to Source DFSTUD
Source to Destination DFSLUS
Destination to Source DFSLUD
Involving the Source of the connection DFMTOS
Involving the Destination of the connection DFMTOD
Involving the Source of the connection DFMCOS
Involving the Destination of the connection DFMCOD

Figure 1: The feature classification schema, and the naming conventions.

Furthermore, not all the features have the same tuning variables; thus, for identifying the set of parameters that needs to be tuned, the features are grouped based on their underlying implementation, not based on their semantic definition and meaning. Consequently, we have *basic features* (i.e., *BF*), *time-based features* (i.e.,*DFST*, *DFMT*), and *connection-based features* (i.e.,*DFSL*, *DFMC*) as the primary types of features that present different tuning parameters. Since these parameters heavily influence the effectiveness of each feature in detecting attacks, it is of great importance to study multiple tuning values for each feature while evaluating its performance. The *BF* category consists of all features that can be extracted from a single packet without requiring any kind of extra information. The feature candidates for this category can be any field of the datagram such as protocol, source and destination ports, flags, ICMP type. Extracting these features is extremely easy and fast since the feature constructor needs to examine only a single packet at a time. No extra dependency information is required, and thus, there is no need for tuning in this case. Even though this category is the easiest type of features that can be created, it is also the most inefficient one to use. On the other hand, the *time-based* and *connection-based* features depend on several tuning factors that naturally lead to different performance values for these features. As explained in order to evaluate each feature we consider a set of tunings that will allow us to decide if that particular feature is or is not reliable in attack detection. The logics behind our proposed tunings are explained in the next two subsections.

3. FEATURE PERFORMANCE EVALUATION

The overall logical architecture view of the proposed evaluation model is depicted in Figure 2. The proposed feature evaluation model combines statistical with fuzzy logic techniques to mathematically extract the usefulness of a feature f_i, in detecting a certain type of attack ξ_m. Our model proposes a solution for the *subset evaluation* step of the feature selection process. In particular, we propose a new *feature dependency measure* for *independent evaluation criteria* that is, to our knowledge, a pioneer method designed for intrusion detection. The method is primarily designed for network-based features, but can be easily extendable to application-based features. Let us define $P\ (f_i|\xi_m)$ as the probability of feature f_i to detect the ξ_m attack category. The higher this value is, the more suitable feature *fi* is for the detection of attack category ξ_m. The proposed evaluation model assumes that the selected features have been already extracted and that the necessary statistical data for the evaluation process has been mined too. For this purpose we use a *Framework for Real Time Network Feature Construction* combined with a *Statistical Profiler Module*.

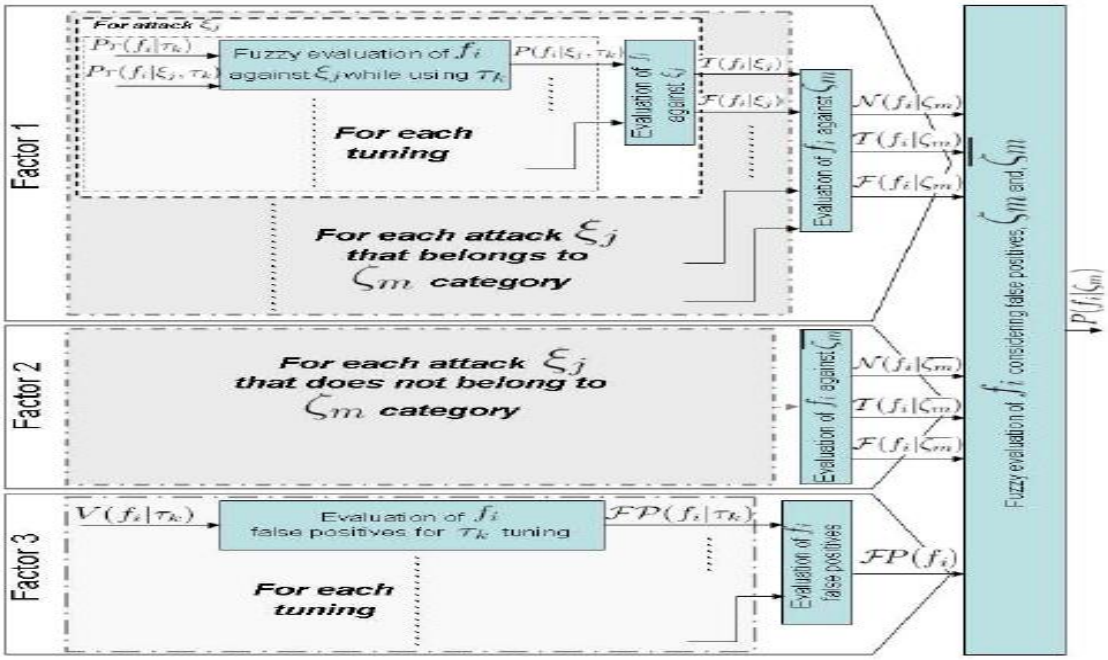
Factor 1
For attack ξ_j
$Pr(f_i|\tau_k)$
$Pr(f_i|\xi_j,\tau_k)$
Fuzzy evaluation of f_i against ξ_j while using τ_k
$P(f_i|\xi_j,\tau_k)$
Evaluation of f_i against ξ_j
For each tuning
$\mathcal{T}(f_i|\xi_j)$
$\mathcal{F}(f_i|\xi_j)$
Evaluation of f_i against ζ_m
For each attack ξ_j that belongs to ζ_m category
$\mathcal{N}(f_i|\zeta_m)$
$\mathcal{T}(f_i|\zeta_m)$
$\mathcal{F}(f_i|\zeta_m)$
Factor 2
For each attack ξ_j that does not belong to ζ_m category
Evaluation of f_i against $\overline{\zeta_m}$
$\mathcal{N}(f_i|\overline{\zeta_m})$
$\mathcal{T}(f_i|\overline{\zeta_m})$
$\mathcal{F}(f_i|\overline{\zeta_m})$
Factor 3
$V(f_i|\tau_k)$
Evaluation of f_i false positives for τ_k tuning
$\mathcal{FP}(f_i|\tau_k)$
For each tuning
Evaluation of f_i false positives
$\mathcal{FP}(f_i)$
Fuzzy evaluation of f_i considering false positives, ζ_m and $\overline{\zeta_m}$
$P(f_i|\zeta_m)$

Figure 2: The overall view of the logical architecture of the proposed evaluation model.

We choose not to include their description here (even though these steps need to be implemented) as they are not part of the feature evaluation process. Similarly, there are other data preprocessing steps that are not mentioned here, but are described in next paper. There are three main input types that the *Feature Performance Evaluation* requires as follows: the feature profile during normal operation, the feature profile during an attack, and the feature values during the normal operation. Let $P_r(f_i|\tau_k)$ represent the feature f_i profile during the normal operation while using the τ_k tuning. Similarly, let $P_r(f_i|\xi_j,\tau_k)$ represent the feature f_i profile during the intrusive stage of attack ξ_j while using the τ_k tuning. Finally, let $V\ (f_i|\tau_k)$ represent the set of all normal values that feature f_i has while using the τ_k tuning.

Let μ_N, σ_N represent the mean and standard deviation of feature *fi* during the normal operation, and while configured using τk tuning value. Similarly, let μ_I, and σ_I and represent the mean and standard deviation of feature *fi* during the ξj attack, and while configured using τk tuning value. The $P_r(f_i|\tau_k)$ will consist of $< \mu_N, \sigma_N >$ tuple while $P_r(f_i|\xi_j\ ;\ b_{\tau k})$ will consist of $< \mu_I\ ,\ \sigma_I >$ tuple.

4. CONCLUSIONS

It consists of three main parts. In the first part a Feature Classification Schema is presented that is meant to impose a single centralized standard over the various types of features that can be extracted from the network. Furthermore, this step also allows our experimental results to include a comprehensive set of features, which contains features from all the presented feature categories. Next, we identified a set of tuning parameters that directly influence the values that each feature will produce, which will inevitably influence their detection performance. Finally, we describe our proposed feature evaluation schema explaining the reasoning behind its design. This evaluation schema mines the intrinsic characteristics of each of the studied features using a hybrid method that employs statistical and fuzzy logic techniques, for producing the final outcome.

REFERENCES

[1] K. Das, *The development of stealthy attacks to evaluate intrusion detection systems*, Master's thesis, MIT Department of Electrical Engineering and Computer Science, June 2000.

[2] M. Dash, K. Choi, P. Scheuermann, and H. Liu, *Feature selection for clustering - a filter solution*, Data Mining, 2002. ICDM 2002. Proceedings. 2002 IEEE International Conference on (2002), 115{122.

[3] M. Dash and H. Liu, *Feature selection for clustering*, PADKK '00: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications (London, UK), Springer-Verlag, 2000, pp. 110{121.

[4] M. Dash, H. Liu, and J. Yao, *Dimensionality reduction of unsupervised data*, Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on, no. 3-8, November 1997, pp. 532{539.

[5] P.A. Devijver and J. Kittler, *Pattern recognition a statistical approach*, Prentice Hall International, 1982.

[6] J. Doak, *An evaluation of feature selection methods and their application to computer security*, Tech. Report CSE-92-18, University of California at Davis, 1992.

[7] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan, *Data mining for network intrusion detection*, Proceedings of NSF Workshop on Next Generation Data Mining (Baltimore, MD), November 2002, pp. 21{30.

[8] P. Domingos, *Context-sensitive feature selection for lazy learners*, (1997), 227{253.

[9] J. G. Dy and C. E. Brodley, *Feature subset selection and order identification for unsupervised learning*, ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning (San Francisco, CA, USA), Morgan Kaufmann Publishers Inc., 2000, pp. 247{254.

[10] L. Ertoz, E. Eilertson, A. Lazarevic, P.N. Tan, P. Dokas, V. Kumar, and J. Srivastava, *Detection of novel network attacks using data mining*, In ICDM Workshop on Data Mining for Computer Security (DMSEC) (Melbourne, FL), Nov. 19 2003, pp. 30{39.

[11] Chapman Flack and Mikhail J. Atallah, *Better logging through formality applying formal specification techniques to improve audit logs and log consumers*, Proceedings of Recent Advances in Intrusion Detection, 3rd International Symposium, (RAID 2000) (Toulouse, France) (H. Debar, L. M, and S.F. Wu, eds.), Lecture Notes in Computer Science, Springer-Verlag Heidelberg, October 2000, pp. 1{16.

[12] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, *A sense of self for unix processes*, Proceedings of the 1996 IEEE Symposium on Security and Privacy (Los Alamitos, CA), IEEE Computer Society Press, 1996, p. 120128.

[13] J.M. Garibaldi and R.I. John, *Choosing membership functions of linguistic terms*, Proceedings 2003 IEEE International Conference on Fuzzy Systems, 2003, pp. 578 { 583.

[14] Anup K. Ghosh, Christoph Michael, and Michael Schatz, *A real-time intrusion detection system based on learning program behavior*, Proceedings of Recent Advances in Intrusion Detection, 3rd International Symposium, (RAID 2000)(Toulouse, France) (H. Debar, L. M, and S.F. Wu, eds.), Lecture Notes in Computer Science, Springer-Verlag Heidelberg, October 2000, pp. 93{109.