Network Security Framework

Kapil Kumar Gupta *Baikunth Nath (Sr. Member IEEE)* *Kotagiri Ramamohanarao*

Department of Computer Science and Software Engineering,
National ICT Australia,
The University of Melbourne
Victoria, Australia 3010.

Summary

In this paper we present a general framework for an Intrusion Detection System which we call as the Layer Based Intrusion Detection System (LBIDS). We base our framework on the fact that any network needs to ensure the confidentiality, integrity and availability of data and/or services which can be compromised only sequentially one after the other, i.e. availability followed by authentication and authorization and finally leading to loss of confidentiality and integrity. Our framework examines different attributes at different layers to effectively identify any breach of security at every layer. This has the advantage that we can effectively divide the computation into smaller parts and if at any stage/layer the system makes a decision that there is an attack, it can simply block that intrusion and save the higher layers from performing any further computation, rather than making a decision by aggregating entire data at a single point as is commonly used in any well known Intrusion Detection System.

Key words:

Network Security, Intrusion Detection, Layered Approach

Introduction

The current state of network is vulnerable; they are prone to increasing number of attacks. These attacks are seldom previously seen and being different they are very hard to detect before subsequent damage is done [15]. Thus securing a network from unwanted malicious traffic is of prime concern. A computer network is more than a group of connected nodes. On one hand it needs to provide continuous services, such as e-mail, to a number of users, while on the other it stores huge amount of data which is of vital significance.

Intrusion Detection techniques employed to detect attacks are now not new. However, Intrusion Detection until recently has been employed for perimeter security and detecting attacks which were targeted towards denial of service or a resource. Recently, there has been increasing concern over safeguarding the vast amount of data stored in a network from malicious modifications and disclosure to unauthorized individuals. The nature of data stored in a network may range from personal information including identity related details, medical history, bank account and credit card details etc. to a company's official details and management plans. Any misuse of this critical data stored in the repositories might lead to drastic consequences. Thus, a network must ensure security of both, the services it provides and the large amount of data that it stores. Hence it is the confidentiality, integrity and availability (CIA) of service and data that needs to be ensured to ensure complete network security.

Intrusion Detection Systems (IDS) are based on two concepts; matching of the previously seen and hence known anomalous patterns from an internal database of signatures or building profiles based on normal data and detecting deviations from the expected behaviour. The first approach is called as Misuse Detection and leads us towards Signature Based IDS while the second is called as Anomaly Detection and leads us to Behaviour based IDS. The Signature based systems though have very high detection accuracy but they fail when an attack is previously unseen. On the other hand, Behaviour based IDS may have the ability to detect new unseen attacks but have the problem of low detection accuracy [11], [18], [14].

Based on the mode of deployment the Intrusion Detection Systems are classified as Network based, Host based and Application based. Network based systems make a decision by analyzing the network logs and packet headers from the incoming and outgoing packet since they are deployed at the periphery of the network. Though they are easy to manage and give a centralized control, they have to work with limited information and are further

Manuscript received July 25, 2006.
Manuscript revised July 30 , 2006.

constrained in case of encryption and network address translation. Host based systems monitor's individual systems and uses system logs extensively to make any decision. However, they are platform dependent and needs to be monitored at each node separately where they are deployed. [7] provides further comparison of the Network based IDS and Host based IDS. In order to bridge the gap between the detection capabilities of a Network based IDS (NIDS) and a Host based IDS (HIDS), use of both NIDS and HIDS is recommended in practical situations [6]. Further, Application based IDS uses application logs as their data sources and can provide maximum security. Since all the modes of deployment of IDS differ in their input they have different detection capabilities. Finally, the Distributed Intrusion Detection Systems (DIDS) are also possible which can be of two types. They are either a cooperative approach between the HIDS, NIDS and a central server where the central server makes any decision on the information provided by the cooperating IDS or are a group of stand alone separate IDS alerting neighbors when one IDS discovers any attack, hence sharing the knowledge of attack within the entire network. Both the approaches in the DIDS suffer from some problems. The first type of DIDS have the problem that the central server is a single point that makes any decision for every cooperating system and hence it has lot of data to be processed and is very difficult to be online. While, in the second type of DIDS discussed above, each system (NIDS or HIDS) is a separate identity and simply tells its neighbor that it discovered an attack. It is well known that to secure a network, we should use a combination of NIDS, HIDS or a DIDS, but the question that we are addressing here is; can we have a single system that is strong enough to analyze all the relevant data as well as minimize the amount of computation required and still is highly accurate in terms of detection accuracy.

The paper is organized as follows: In Section 2 we discuss the motivation for our framework followed by related work in Section 3. We discuss our proposed framework in Section 4 and finally conclude in Section 5.

2. Motivation for our Framework

Current systems consider availability, privacy (or confidentiality) and integrity in isolation of each other. However, we believe that all the three are related and hence can not be treated as separate problems. Based on this we introduce the concept of LBIDS, which is not a group of IDS cooperating together as in case of a Distributed IDS, but is a self contained single system which tries to identify the anomalies by a series of tests in succession. This would have the advantage of reducing the computation and increasing the detection accuracy. This is attributed to the fact that once an anomaly is detected at a layer, it saves the computation required by subsequent layer(s) by simply blocking it at the point of identification. Detection accuracy can be increased as the features that are selected to be evaluated to make any decision at a particular layer are optimized to detect that particular attack category. Hence, on one hand, this gives us the flexibility to include a large number of features and, on the other; it helps to divide these features into few groups or layers so that different features can be used at different layers. Further, we can have a single feature that is significant for detection in more than one layer (which also helps to preserver the correlation between two layers). Since we address the three basic security features together, our framework can address any type of attack category including the unknown or undiscovered attacks. Such a system is essentially based on anomaly detection and is an Application based IDS. This is because it is only at the application level, when any semantic information can be obtained from any packet. However, since our system is progressive in nature, it can also be used as a Network or Host based IDS.

3. Related Work

Large amount of work has been done in the area of intrusion detection and a number of techniques including *data mining approaches, clustering, naive Bayesian classifiers, Bayesian networks, hidden Markov models, decision trees, artificial neural networks, support vector machines, genetic algorithm, agent based approaches* and many others have been described in order to detect intrusion. We describe these techniques here particularly with regards to the data they analyze before they label any event as intrusion.

Data mining based approaches for Intrusion Detection are based on building classifiers based on discovering relevant patterns of program and user behaviour. Association rules [8] and frequent episodes are used to learn the record patterns that describe user behaviour [20], [21]. Data mining

approaches can deal with symbolic data and the features can be defined in the from packet and connection details. Thus, mining of features is limited to entry level of the packet and also requires the number of attributes to be large and the records are sparsely populated, otherwise they tend to produce very large number of rules which increases the complexity [5]. Clustering of data has been applied extensively for intrusion detection using various clustering methods including k-means, fuzzy c-means and many others [23], [25]. However, one of the main drawbacks of clustering techniques is that it is based on calculating the distance between the observations and hence the attributes of the observations must be numeric. Symbolic attributes can not be used for clustering which results in inaccuracy. Naive Bayes classifiers are also proposed in [9], however, they make very strict independence assumption between the attributes [26]. In [17] the Bayesian network is used to remove the threshold and combine the results of individual models to reach a final result. However, they tend to be attack specific and build a decision network based on special features of each attack. Thus, the size of the Bayesian network increases rapidly as the number of features considered increases and the type of attacks modeled increases. Hidden Markov models have also been used in intrusion detection. [28], [12], [27] describes the use of hidden Markov models for modeling the normal sequence of system calls [13] of a privileged process, which can then be used to detect anomalous traces of sequence calls. However, modeling the system calls alone may not always provide accurate classification as in such cases various connection level features are ignored. Further, hidden Markov models are generative models and fail to model long range dependency between the observations. Decision trees [9] have also been used for intrusion detection. The problem with the decision trees is to select the best attribute for each decision node during the construction of the tree. One such criterion is to use the gain ratio as in C4.5. The decision trees suffer from similar problems as the Bayesian networks. The decision trees tend to grow in size and complexity as the number of attributes increases. Decision trees can be easily used for building the misuse detection systems, but, it is very difficult to construct anomaly detection system using decision trees. [10], [24], [22], [29] discusses the use of Artificial Neural Networks for network intrusion detection. Though the neural networks can work effectively with noisy data but they require large amount of data during training and it is often hard to select the best possible neural network architecture. Support Vector Machines (SVM) which maps real valued input feature vector to higher dimensional feature space through non-linear mapping have been used for detecting intrusions in [22]. The SVM's provide real time detection capability and can deal with large dimensionality of data. However, they are used effectively for binary class classification only. Along with these, other techniques for detecting intrusion includes the use of genetic algorithms and agent based approach including autonomous agents for intrusion detection [2] and probabilistic agent based approach for intrusion detection [3] which are generally aimed at a distributed Intrusion detection system.

The 1999 KDD intrusion detection data set, which is a version of the 1998 DARPA intrusion detection data set prepared and managed by the MIT Lincoln lab, and the system call data set collected at the University of New Mexico have been widely used to report various experimental results on intrusion detection. The DARPA data set presents data as a collection of records where each record presents a summary of a connection or sequence of packets between a specific source and target IP address at certain well defined times [1], while the system call data is the traces of system calls generated by certain selected routines such as sendmail where each trace is just a sequence of system call and its corresponding process id [4].

All the above mentioned techniques for detecting intrusions are primarily targeted at ensuring availability. There are methods in [30], [16], [19] which are meant to ensure confidentiality and integrity of the data stored in databases. They use the database logs either to build the normal user profiles [16], or to extract signatures for detecting known attacks as discussed in [19].

However, to ensure that a network is secure, we need to provide confidentiality and integrity along with availability. As we discuss in the next section, our framework aims at providing all the three (confidentiality, integrity and availability) together in a single system.

4. Our Framework

As discussed in Section 3, either the current systems suffer from a number of drawbacks in terms of detection capability and accuracy or they

are highly specific to addressing a single issue in security. Hence, with the current setup entire network security is far from reachable. We propose a framework for intrusion detection which we call as the LBIDS.

To ensure complete network security i.e. to provide confidentiality, integrity and availability (CIA), we need a system which is both specific in detecting attacks targeted individually at the CIA by selecting only a small set of features which are significant to detection for a particular category, as well as is capable to correlate the results to ensure complete network security. The system not only needs to perform this task with high accuracy but also needs to do it at a stage as early as possible as it reduces the effect of the attack and also reduces the computation required by the system.

Our system is based on the fact that attacks targeted at confidentiality, integrity and availability can be detected individually by selecting different attributes for each of the three. Further, the complexity of the system or the number of features that are significant for detecting attacks for any higher layer may be more than its previous layer as the higher layer can also involve features which are present in the previous layer. For example, in detecting a DoS attack, i.e. in ensuring availability, we might not be interested in the finding out which file was accessed, while this becomes significant when we want to ensure data integrity and privacy. Further, when ensuring data privacy and integrity, we are not only interested in finding which file was accessed but we also need to take care of the user permissions and access pattern of any user or group. Hence, since the very nature of the CIA is different we need to evaluate different set of features to effectively discover attacks at different layers. Further, it seems very logical that a file cannot be modified unless it is available. We also believe that availability, confidentiality and integrity can be compromised only sequentially, though it is not necessary for one to be a pre-requisite for the other, i.e. confidentiality can be attacked even though there is no attack on availability. Thus, before a file is read or written, we make sure that it is free from a DoS attack or attack on confidentiality.

In order to label an event as normal or as an attack the current intrusion detection systems either reduce the number of features considered to make any decision, thus compromising the detection capability of the system, or make use of many such features making the system very complex and non-incremental.

To reduce system complexity and to make the system incremental, i.e. making the system respond to whatever information is available at the current instant, and thus avoiding any decision making only at the last stage, we propose a layered intrusion detection system. Hence, we split our system into a number of sequential overlapping layers, which we discuss next, where each layer evaluates certain specific features which are significant for detecting attacks targeted at that particular layer. Since, we have divided our system into a number of layers, each layer can result in detecting attacks with high accuracy as it considers all the features necessary for detection at that layer and at the same time it requires minimum effort as the over all detection is divided between number of layers. Further, each next layer in the system uses a set of features which is a combination of some selected features from the previous layer, though it leads to redundancy but is required to link the adjacent layers, and other unique features which are significant to that particular layer.

4.1 Description of Layers

We propose a three layer system to ensure complete security viz. availability, confidentiality and integrity, each layer corresponding to one aspect of security. The layers are sequential and overlapping i.e. layer one followed by layer two followed by layer three, where each layer has some unique features and some features from its previous layers. This ensures that each layer is stand alone and is able to effectively block the type of intrusion which it is meant to block. Sharing of some features from previous layers is necessary to ensure that the layers are linked together. This is important because as we move to any higher layer, various semantic features needs to be related to the non-semantic features such as connection features to ensure better detection capabilities.

In our proposed framework, the first layer or the connection establishment layer corresponds to the packet level features such as source and destination IP address, number of connections to the host, source and destination port number, user ID etc. and is optimized to detect attacks exploiting the availability aspect such as DoS attacks, probes, etc. The second layer which is the privacy layer ensures data confidentiality and refers to features such as files accessed, data retrieved etc. The third

layer or the access control layer ensures integrity of data and is more concerned with the file modifications, user privileges etc. and is also the last layer in our proposed architecture of network security. It is worth mentioning that the access pattern or the privacy layer itself requires some packet level features used in the first layer and so does the access control layer. We represent this layered architecture in Figure 1.

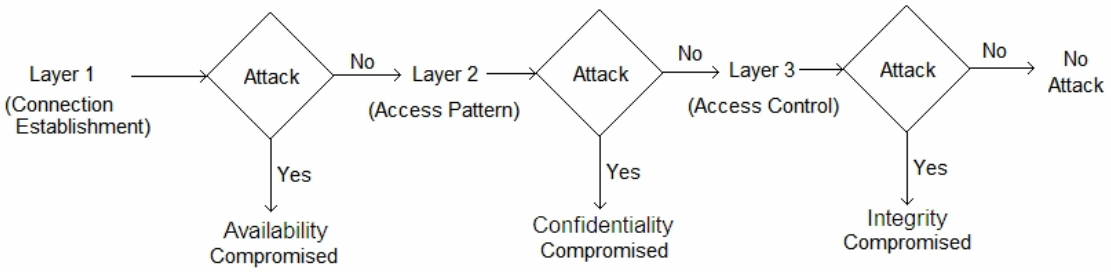

Fig. 1 Layered Approach

4.2 Comparison of our framework

We compare our framework with other current network security frameworks. The task of network security has been mainly confined to the availability aspect, but there are systems that have been implemented to ensure confidentiality, integrity and availability. However, they treat the three security aspects individually and thus results in separate systems for each. In contrast, we consider the three aspects as highly related and propose a single unified framework. Also, the current frameworks, such as the common intrusion detection framework architecture [31], club a number of standalone intrusion detection systems which requires all the standalone systems to understand common language and semantics to interoperate. Our framework on the other hand is based on the three layers, each of which is modeled to ensure availability, confidentiality and integrity sequentially. Our framework is based on ensuring what needs to be preserved rather than protecting from different and unknown kind of attacks.

Our framework has the advantage that it is not specific to any particular type or group of attack as we address the three basic features of security viz. confidentiality, integrity and availability and bind them together in a single system rather than creating different system for ensuring each security aspect. As already discussed our proposed framework is less computational expensive and is incremental to the amount of data that is analyzed, thus, making the approach online, feasible and highly flexible. Further, since the system makes a series of decisions by grouping various layer specific features together our system can be customized for any specific application and can also be used as a stand alone Network or Host based system. Additionally, we can make use of any of the available technique, as discussed in Section 3, for building an effective intrusion detection system. Our framework essentially provides a method that can help to reduce the complexity of the system by simply dividing the task into a sequence of tasks based on the three basic security concept.

5. Conclusions and Future Work

In this paper we proposed a simple yet practical layered approach to intrusion detection/prevention and discussed its various advantages with regards to accuracy and computation. We discussed that such a system would be less computational intensive and more accurate. We are currently evaluating different layers individually and as part of our future work, we plan to implement this framework as a single system. We believe in prevention over cure, as cure may or may not be achieved.

Acknowledgments

We thank National ICT Australia, Victoria Lab for funding and supporting this research.

References

[1] http://kdd.ics.uci.edu//databases/kddcup99/kddcup99.html. KDD Cup 1999 Data: (Last assessed: July 02 2006).

[2] http://www.cerias.purdue.edu/research/aafid/, autonomous agents for intrusion detection. Online article (Last assessed: July 12 2006).

[3] http://www.cse.sc.edu/research/isl/agentIDS.shtml, probabilistic agent based approach for intrusion detection. Online article (Last assessed: July 06 2006).

[4] http://www.cs.unm.edu/˜immsec/systemcalls.htm. Computer Immune Systems: (Last assessed: July 02 2006).

[5] http://www.dsto.defence.gov.au/publications/2345/DSTO-GD-0286.pdf. Online article (Last assessed: July 06 2006).

[6] http://www.sans.org/resources/idfaq/host based.php. Online article (Last assessed: July 06 2006).

[7] http://www.windowsecurity.com/articles/Hids vs Nids Part1.html. Online article: (Last assessed: July 06 2006).

[8] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proceedings of the 1993 International Conference on Management of Data (SIGMOD 93), ACM Press, vol(22) Issue 2, 1993, pages 207–216.

[9] N. B. Amor, S. Benferhat, and Z. Elouedi. Naive bayes vs decision trees in intrusion detection systems. In Proceedings of the ACM symposium on Applied computing, ACM Press, 2004, pages 420–424.

[10] H. Debar, M. Becke, and D. Siboni. A neural network component for an intrusion detection system. In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, 1992, pages 240–250.

[11] D. Denning. An intrusion-detection model. IEEE Transactions on Software Engineering, vol. (SE-13), no. (2), 1987, pages 222–232.

[12] Y. Du, H. Wang, and Y. Pang. A hidden markov models-based anomaly intrusion detection method. In Fifth World Congress on Intelligent Control and Automation, 2004, (WCICA ’04), IEEE Press, vol. (5), 2004, pages 4348–4351.

[13] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for UNIX processes. In Proceedinges of the IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, 1996, pages 120–128.

[14] K. Ghosh. Learning program behavior profiles for intrusion detection. In Proceedings of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring, 1999, pages 51–62.

[15] K. K. Gupta, B. Nath, K. Rao, and A. Kazi. Attacking confidentiality: An agent based approach. In Proceedings of IEEE International Conference on Intelligence and Security Informatics, Lecture Notes in Computer Science, Springer Verlag, vol. (3975), 2006, pages 285–296.

[16] Y. Hu and B. Panda. A data mining approach for database intrusion detection. In Proceedings of the 2004 ACM symposium on Applied Computing, ACM press, pages 711–716.

[17] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur. Bayesian event classification for intrusion detection. In 19th Annual Computer Security Applications Conference, 2003, IEEE Computer Society, pages 14–23.

[18] S. Kumar and E. H. Spafford. An application of pattern matching in intrusion detection. In Technical Report CSDTR-94-013, Purdue University, 1994, pages 94–113.

[19] S. Y. Lee, W. L. Low, and P. Y. Wong. Learning fingerprints for a database intrusion detection system. In 7th European Symposium on Research in Computer Security, ESORICS, Lecture Notes in Computer Science, Springer-Verlag, vol. (2502), 2002, pages 264–279.

[20] W. Lee and S. Stolfo. Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, 1998, pages 79–94.

[21] W. Lee, S. Stolfo, and K. Mok. Mining audit data to build intrusion detection models. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1998, pages 66–72.

[22] S. Mukkamala, G. Janoski, and A. Sung. Intrusion detection using neural networks and support vectormachines. In Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN ’02, IEEE Press, pages 1702–1707.

[23] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In ACM Workshop on Data Mining Applied to Security (DMSA 2001). ACM Press, 2001.

[24] J. Ryan, M. J. Lin, and R. Miikkulainen. Intrusion detection with neural networks. In Advances in Neural Information Processing Systems. MIT Press, 1998.

[25] H. Shah, J. Undercoffer, and A. Joshi. Fuzzy clustering for intrusion detection. In The 12th IEEE International Conference on Fuzzy Systems (FUZZ ’03), pages 1274–1278. IEEE Press, vol. (2) 2003.

[26] Valdes and K. Skinner. Adaptive, model-based monitoring for cyber attack detection. In Recent Advances in Intrusion Detection (RAID 2000), Lecture Notes in Computer Science, Springer-Verlag, no. (1907), pages 80–92.

[27] W. Wang, X. H. Guan, and X. L. Zhang. Modeling program behaviors by hidden markov models for

intrusion detection. In Proceedings of International Conference on Machine Learning and Cybernetics, IEEE Press, vol. (5), 2004, pages 2830–2835.

[28] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: alternative data models. In Proceedings of the 1999 IEEE Symposium on Security and Privacy, IEEE Press, 1999, pages 133–145.

[29] Z. Zhang, J. Li, C.N. Manikopoulos, J. Jorgenson, and J. Ucles. Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, 2001, pages 85–90.

[30] Y. Zhong, Z. Zhu, and X.L. Qin. A clustering method based on data queries and its application in database intrusion detection. In Proceedings of the fourth International Conference on Machine Learning and Cybernetics, IEEE Press, vol. (4), 2005, pages 2096–2101.

[31] http://gost.isi.edu/cidf, Common Intrusion Detection Framework. (Last assessed: July 20 2006).