Power-Aware Hybrid Intrusion Detection System Using Support Vector Machine in Wireless Ad Hoc Networks

P. KIRAN SREE[†] Dr.I .RAMESH BABU[† †] N.S.S.S.N USHA DEVI[†††]

[†] *Associate Professor, Department of Computer Science, S.R.K Institute of Technology, Enikepadu, Vijayawada,India, pkiransree@gmail.com, Mobile: +919959818274.*

[††]*Head of the Department, Comuter Science, Acharya Nagarjuna University, Guntur.*

[†††] *Graduate Student of C.S.E,J.N.T.University.*

ABSTRACT

Ad hoc wireless network with their changing topology and distributed nature are more prone to intruders. The network monitoring functionality should be in operation as long as the network exists with nil constraints. The efficiency of an Intrusion detection system in the case of an ad hoc network is not only determined by its dynamicity in monitoring but also in its flexibility in utilizing the available power in each of its nodes. In this paper we propose a hybrid intrusion detection system, based on a power level metric for potential ad hoc hosts, which is used to determine the duration for which a particular node can support a network-monitoring node.. Power –aware hybrid intrusion detection system focuses on the available power level in each of the nodes and determines the network monitors. Power awareness in the network results in maintaining power for network monitoring, with monitors changing often, since it is an iterative power-optimal solution to identify nodes for distributed agent-based intrusion detection. The advantage that this approach entails is the inherent flexibility it provides, by means of considering only fewer nodes for re-establishing network monitors. The detection of intrusions in the network is done with the help of support vector machine (SVM). The SVM's classify a packet routed through the network either as normal or an intrusion. The use of SVM's enable in the identification of already occurred intrusions as well as new intrusions.

1. INTRODUCTION

An intrusion is defined as "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource". Intrusions in wireless networks amount to *interception, interruption, or fabrication* of data transmitted across nodes, wherein an intruder node attempts to access unauthorized data.

Hybrid intrusion detection systems are inherently reconfigurable, since the agents can easily be migrated to other hosts, and are by themselves lightweight, and thus suit the power sensitive nature of networks such as wireless sensor networks. We adopt a hierarchical model for Intrusion Detection and extend it to include power awareness of individual nodes. We utilize the power level metric PLANE as described in [1], for comparing power levels across nodes for running agent-based network monitoring processes. A complete analysis of possible network threats to general ad-hoc networks is found in [3, 10].We adopt the hierarchical model proposed in [2] and, modify SPAID [1], extending it to provide efficient power aware solution for dynamic networks.

2. SURVEY OF RELATED WORK

Numerous detection systems have been proposed to tackle the problem of intrusion in wireless networks some of which are an extension of intrusion detection system in wired networks. Few deal with network based IDS [6] and few with host based IDS [7], all which are based on lightweight agents [5, 6, 7]. Power awareness in mobile ad hoc networks [4] becomes a major issue when considering intrusion detection in larger networks.

3. PHIDS

The agent-based model proposed in [10] approaches the IDS problem with a technique that handles intrusions with an agent running on each system. Further, the model in [10] is not suitable for a power-aware IDS , since such a system warrants energy consumption in systems irrespective of their current battery levels, i.e. it suggests an IDS without considering the feasibility of the assumption that network monitoring and analysis is justified in nodes with minimal power, such as robust wireless sensor networks (WSN).

3.1 Modular IDS Architecture

The IDS we consider is built on a mobile agent framework as in [1]. It is a non-monolithic system and employs several sensor agents that perform certain functions, such as Network monitoring Host monitoring, Action, Decision making.

3.2. Agent Distribution

As a modification over the previous approach for agent distribution [1], the nodes on a wireless ad-hoc network, that are elected as network monitors will include the action and decision making modules. To save resources, some of the functionality must be distributed efficiently to a (small) number of nodes. The decision making module incorporates the energy metric Power Loss/Availability for Network-monitoring Estimate (PLANE), a node-specific measure of the mean power loss per node for running the network monitoring agent. PLANE can directly be related to the wireless protocol used, mean number of wireless links for the specific node, average node maintenance energy consumption, and the battery power remaining.

3.3. Calculating PLANE

The calculation of PLANE involves calculating the duration for which the node can continue to support a network monitor along with its normal operations. We therefore calculate PLANE by calculating the time for which node can last as the network monitoring node as shown below in Equation 1.

$$PLANE = \frac{BPR}{TEC_{nm}} \quad (1)$$

In Equation 1, BPR is the total Battery Power Remaining at the instant of node selection and TECnm is the Total Energy Consumption with network monitoring node processes running. In the absence of measurement of exact networking monitoring energy consumption, we assume PLANE as PLANE'. The value PLANE' is typically available directly from most distributed wireless networks, such as sensor networks, and hence finds a presence in the above calculation.

$$PLANE' = \frac{BPR}{TEC} \quad (2)$$

3.4. The PHIDS Algorithm

The PHIDS algorithm uses the agent hierarchy presented , with a significantly adapted node selection mechanism to incorporate power-awareness, and is best detailed by the following eight steps.

Step 1*: Set PLANE threshold*. Set a constraint on the PLANE value of nodes which are allowed to compete for becoming a network monitoring node.
Step 2*: PLANE Calculation and PLANE Ordered List (POL).* Arrange the different nodes in increasing values of PLANE as calculated previously, for all nodes which satisfy the PLANE Constraint.
Step 3*: Hop Radius.* Set the hop radius to one initially, and increment for each insufficient node selection with the current hop radius.
Step 4*: Expand Working Set of Nodes.* Consider node selection incrementally, initially from the first node, (node with highest PLANE), to finally the set of all nodes in the network, incrementing the set of nodes under consideration by one node each time. We call this set the working set (WS) of nodes. The WS is expanded only if the addition leads to an increase in number of represented nodes.
Step 5*: Voting. The voting scheme* for Node Selection, is similar to that in [2], except that we limit the candidates to just the nodes which are part of WS.
Step 6*: Check acceptability of nodes*. If all links/nodes are not represented by the set of nodes covered by the voting scheme, then we expand the WS and repeat the process from Step 4. If WS equals the POL, then increment the hop radius, and repeat from Step 3. It is suggested that the increment in hop radius be considered a final resort, as it effectively increases the amount of processing per monitoring node.
Step 7*: Cluster Setup.* Set individual clusters with the nodes in the working set as root and the nodes being monitored by it as child nodes.
Step 8*: Re-run.* Changes in power levels of the root nodes in each cluster will be signaled to the child and the vote count as in step 5 takes place within the cluster to form a new monitoring node.

Steps 1 to 6 are similar to that of SPAID but are suitable even to highly mobile networks, since decision making module does not rest with very few nodes and only clusters are altered on each addition of a new node.

4. INTRUSION DETECTION MODULE USING SVM:

Support Vector Machines (SVM's) represent a generalized linear classifier that looks for the maximum margin hyper plane between two classes in the feature space i.e. the SVM's try to find a hyper plane between two classes which maximizes the minimum distance between the hyper plane and the data points.

4.1 Maximum Margin Hyper plane

The optimal position of the class boundary is obtained as a linear combination of some training samples that are placed near the boundary itself, and are called support vectors.

The hyper planes are defined by the linear set of equations given by:

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \quad \mathbf{w} \in \mathbb{R}^d, \quad b \in \mathbb{R}$$

Each input x is subject to the decision function O(x):

$$O(x) = \text{sign}(w.x + b)$$

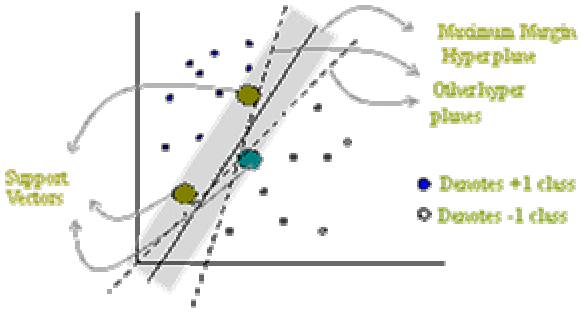
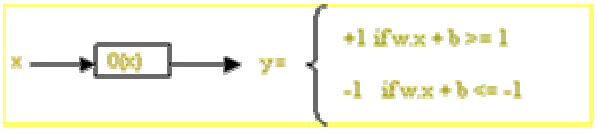


Fig 1: SVM as a Linear Classifier



The margin width of the hyper plane is computed by considering the plus plane and minus plane

Plus plane = {x: w.x +b = +1}
Minus plane = {x: w.x +b = -1}

The perpendicular distance of each of the plus and the minus planes respectively from the classifier boundary are given $\frac{|1-b|}{||w||}$ and $\frac{|-1-b|}{||w||}$.Hence the margin width is $\frac{|1-b+1+b|}{||w||} = \frac{2}{||w||}$ and the pair of hyper planes that gives the maximum margin can be found by minimizing w2..This is a quadratic programming (QP) problem. It is solved by LaGrange Formulation of the margin width equation. We have seen an overview of how to use SVM for binary classification purpose.

4.2 Details of Algorithm

There are two phases to the implementation of classification problem using SVM's. They are the

- Training or Learning phase
- Testing or Recognition phase

A detailed description of the algorithm is shown in the figure below. It has the two phases-training phase and testing.

4.3 Training or Learning phase

In the training phase the optimal hyper planes for each of the binary classifiers are constructed based on the training set data. The system is trained with a lot of sample intrusions and their attributes. These images form the basis in identifying the image the user queries. The attribute vector for each of these intrusions is found and is stored in a feature database. The instances belonging to a similar class are grouped into a single category. There are 2 classes namely INTRUSION class and NORMAL class, therefore build 2 SVM's. Each SVM is trained to identify its class by estimating its optimal hyper plane for each SVM.

4.4 Testing or Recognition phase

In the testing phase the attributes of the instance are used to query. The attribute vector is calculated for the image and is given as input to the pool of trained.

SVM's which identifies the class to which the instance belongs and takes the necessary steps.

5. PERFORMANCE COMPARISON:

Evaluating the extended algorithm, PHIDS, in terms of power, results in much better utilization of the available power.

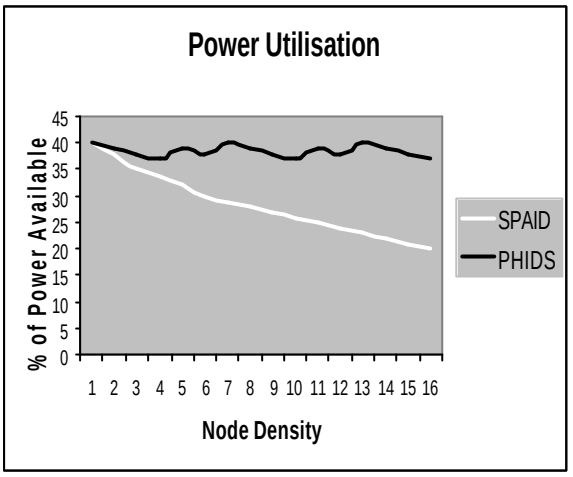


Fig.2. Performance Comparison – SPAID vs. PHIDS

Splitting up of larger networks to clusters and manipulating power levels and thresholds within them provides a power optimal solution than that of SPAID, which requires the entire network.
Also SPAID [1] was considered only for minimally mobile networks with increments in hop radius for more dense networks, while PHIDS with tree based clusters can prove efficient even in the case of dynamic networks.

6. CONCLUDING REMARKS

In this paper, sufficient base has been provided to understand the efficient functioning of the PHIDS Algorithm in determining the duration for which a node can support a network monitoring function in wireless ad hoc networks. The preliminary results show that the PHIDS Algorithm gives good results on sparse as well as dense mobile networks. As it is evident from the power utilization performance evaluation the PHIDS algorithm proves to be scalable and even more efficient as the number of nodes increases i.e. as the size of the wireless ad hoc network increases. Re-run over the entire network for node selection needs to be done often only with changing network topologies.

REFERENCES

1. T. Srinivasan, Jayesh Seshadri, J.B. Siddharth Jonathan,and Arvind Chandrasekhar, (2005). 'A System for Power-Aware Agent-Based Intrusion Detection (SPAID) in Wireless Ad Hoc Networks', Springer-Verlag Berlin Heidelberg.
2. Kachirski, O. and Guha, R.: (2003) Efficient Intrusion Detection using Multiple Sensors in Wireless Ad Hoc Networks", 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 2
3. Zhou, L., and Haas, Z.J. (1999)*: Securing Ad Hoc Networks, IEEE Networks Special Issue onNetwork Security. November.*
4. Ahmed M. Safwat, Hossam S. Hassanein, and Hussein T. Mouftah (2002), *Handbook of Ad hoc Wireless Networks, CRC Press, Dec. 2002,* 'Power-Aware Wireless Mobile Ad hoc Networks'
5. D. Dasgupta and H. Brian, (2001) "Mobile Security Agents for Network Traffic Analysis", Proceedings of DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01, Volume: 2, pp. 332–340.
6. G. Helmer, J. Wong, V. Honavar, L, Miller, (2000) "Lightweight Agents for Intrusion Detection", Technical Report, Dept. of Computer Science, Iowa State University
7. M.C. Bernardes and E. Santos Moreira, (2000). "Implementation of an Intrusion Detection System based on Mobile Agents", Proceedings of International Symposium on Software Engineering for Parallel and Distributed Systems, pp. 158-164.
8. Tao, J., Ji-ren, L., and Yang, Q.: (2000) "The Research on Dynamic Self-Adaptive Network Security Model Based on Mobile Agent", Proceedings of 36th International Conference on Technology of Object-Oriented Languages and Systems
9. Bernardes, M.C., and Moreira, E.S.: (2000). "Implementation of an Intrusion Detection System based on Mobile Agents", Proceedings of International Symposium on Software Engineering for Parallel and Distributed Systems, pp. 158-164
10. Zhang, Y., and Lee, W.: (2000). "Intrusion Detection in Wireless Ad-Hoc Networks", Proceedings of the 6th Annual International

Conference on Mobile Computing and Networking, MobiCom, pp. 275-283

P.KIRAN SREE received his **B.Tech** in Computer Science & Engineering, from J.N.T.U and **M.E** in Computer Science & Engineering from Anna University. He has published many technical papers; both in international and national Journals .His areas of interests include Parallel Algorithms, Artificial Intelligence, Compile Design and Computer Networks. He also wrote books on Analysis of Algorithms, Theory of Computation and Artificial Intelligence. He was the reviewer for many IEEE Society Conferences in Artificial Intelligence and Networks. He was also member in many International Technical Committees. He was the technical editor for Journal of Artificial Intelligence, Journal of Software Engineering, Research Journal of Information Technology, and Information Technology Journal. He is now associated with S.R.K Institute of Technology, Vijayawada.