Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks

Oleg Kachirski
School of Computer Science
University of Central Florida
Orlando, FL, U.S.A.
oleg@cs.ucf.edu

Ratan Guha
School of Computer Science
University of Central Florida
Orlando, FL, U.S.A.
guha@cs.ucf.edu

Abstract

In this paper we propose a distributed intrusion detection system for ad hoc wireless networks based on mobile agent technology. Wireless networks are particularly vulnerable to intrusion, as they operate in open medium, and use cooperative strategies for network communications. By efficiently merging audit data from multiple network sensors, we analyze the entire ad hoc wireless network for intrusions and try to inhibit intrusion attempts. In contrast to many intrusion detection systems designed for wired networks, we implement an efficient and bandwidth-conscious framework that targets intrusion at multiple levels and takes into account distributed nature of ad hoc wireless network management and decision policies.

1. Introduction

With rapid development of wireless network applications, security became one of the major problems that wireless networks face today. While firewalls may prove to be an efficient first line of defense in wired networks, that is certainly not the case in the wireless world. Wireless transmissions are subject to eavesdropping and signal jamming. Physical security of each node is important to maintain integral security of the entire network. Ad hoc wireless networks are totally dependent on collective participation of all nodes in routing of information through the network. These are some of the major problems that wireless networks face today. As the uses of such networks grow, users will demand secure yet efficient, low-latency communications.

Intrusion detection is one of key techniques behind protecting a network against intruders. An Intrusion Detection System is a system that tries to detect and alert on attempted intrusions into a system or network, where an intrusion is considered to be any unauthorized or unwanted activity on that system or network. Extensive research has been done in this field [1, 2, 4, 5, 7, 8] and efficient IDS systems have been designed for wired networks. These systems usually monitor user, system and network-level activities continuously, and normally have a centralized decision-making entity. While these architectures have proven to be effective [1], most of the techniques will not produce expected results when applied to wireless networks, due to some inherent properties that wireless networks possess, as mentioned further.

In this paper, we will concentrate our discussion on ad hoc wireless networks. Ad hoc wireless network is a collection of mobile nodes that establish a communication protocol dynamically. The nodes may join the network at any time and communicate with entire network via the neighboring nodes. There are no base stations, and each member of such a network is responsible for accurate routing of information, and takes part in routing decisions. Due to arbitrary physical configuration of an ad hoc network, there is no central decision making mechanism of any kind – rather, the network employs distributed mechanisms of coordination and management. What really makes a difference between fixed wired and mobile wireless networks is the fact that mobile nodes have a very limited bandwidth and battery power. Network packet monitoring is performed at gateways in a fixed network, but a concept of a gateway in a wireless network is very vague, depending on the type of network and routing algorithms used. Efficient host-based monitoring requires large amounts of CPU processing power, and hence is energy consuming.

Our proposed IDS system takes into account the above considerations to provide a lightweight, low-overhead mechanism based on mobile security agent concept. Essentially, an agent is a small intelligent active object that travels across network to be executed on a certain host, then it returns with results back to the originator. All the decisions, including network traversing, are left to an agent. Agents are dynamically updateable, lightweight, have a specific functionality and can be viewed as components of a flexible and dynamically configurable IDS. These qualities make them a choice for security framework in bandwidth and computation-sensitive

wireless ad hoc networks. We utilize mobile agents at several usage levels and process their response in cluster heads – special nodes that are elected using a distributed algorithm within a cluster. The main contribution of our approach is the efficient distribution of mobile agents with specific IDS tasks according to their functionality across a wireless ad hoc network. The other advantage of our approach is to restrict computation-intensive analysis of overall network security state to a few key nodes. These nodes are dynamically elected, and overall network security is not entirely dependent on any particular node.

2. Previous works

2.1. IDS classification

Traditionally, IDS systems for fixed networks were divided into two categories – network-based and host-based IDS. Network-based systems (NIDS) passively or actively listen on the network, and capture and examine individual packets flowing through a network. In contrast to firewalls, NIDS can analyze the entire packet, not just IP addresses and ports. They are able to look at the payload within a packet, to see which particular host application is being accessed, and with what options, and to raise alerts when an attacker tries to exploit a bug in such code, by detecting known attack signatures. Network IDS are host-independent, and can run as "black box" monitors to cover the entire networks of systems. In practice, active scanning slows down the network considerably, and can effectively analyze a limited bandwidth network. NIDS often required dedicated hosts or special equipment, and thus can be prone to the network attack. A few reliable network-based intrusion detection systems are described in [1, 5, 7, 8].

While network-based IDS look at all the traffic on a network, host-based intrusion detection systems [1, 2, 3, 6] are concerned with what is happening on each individual host. They are able to detect actions such as repeated failed access attempts or changes to critical system files, and normally operate by accessing log files or monitoring real-time system usage. To ensure effective operation, host IDS clients have to be installed on every host on the network, tailored to specific host configuration. Host-based IDS do not depend on network bandwidth, but are used for smaller networks, where each host dedicates processing power towards the task of system monitoring. As mentioned, these systems are host-dependent, and can considerably slow down the hosts that have IDS clients installed.

IDS systems are functionally divided into two classes – anomaly detection and misuse detection systems. Anomaly detection bases its ideas on statistical behavior modeling. Initially, a statistical model is built over time that can be used to accurately predict user behavior based on previous system usage patterns (or a network traffic based on prior traffic patterns). This model detects intrusion detections in a very accurate and consistent way, and has a low level of false alarms, under condition that the system under surveillance follows static behavioral patterns. This class of IDS systems is well suited to detect unknown or previously not encountered attacks. Misuse detection systems monitor networks and hosts for known attack patterns. This class of IDS systems is useful in networks with highly dynamic behavioral patterns, and is a choice of many commercial IDS products. However, a frequently updated (and large) database of known attack signatures should be maintained. Both classes of IDS can be used on host-based and network-based IDS systems.

2.2. Problems related to wireless networks

A number of general problems with IDS systems include high costs due to local management, failure to exhibit scalability, fine-tuning requirements based on specifics of a particular system, need for frequent database updates, and passive behavior (inability to make decisions on type of actions to be undertaken). Little research has been done in the area of IDS systems designed for wireless networks. The structural and behavioral differences between wired and wireless mobile networks make existing IDS designs inapplicable to the wireless networks. As discussed above, wireless networks don't have a fixed, well-protected communication medium – instead, all communication is conducted in an open air environment. This makes it impossible to monitor network traffic at bottlenecks (thus capturing and analyzing majority of packets passing through the network). Therefore, network monitoring in wireless ad hoc networks is performed at every network node. This approach is inefficient in terms of network bandwidth consumption and increased computational power – resources that are highly limited in a wireless network. Host-based monitoring also contributes to a higher amount of processing on each host, thus shortening battery life and slowing down the host. Physical mobile host security is an issue, as each host contains public and private keys used to encrypt information over the network, and if captured, the network is subject to eavesdropping.

Applying functionality-based fixed network IDS models also has limitations. Anomaly detection model is built on a long-term monitoring and classifying of what is a normal or abnormal system behavior. Ad hoc wireless networks are very dynamic in structure, giving rise to apparently random communication patterns, thus making it challenging to build a reliable behavioral model. Misuse detection requires maintenance of an extensive database of attack signatures, which in the case of ad hoc network would have to be replicated among all the hosts. This will

result in an extended initial setup time and decrease in useful computational power of each host.

A few papers have suggested IDS systems targeted at wireless networks. In [3], a distributed IDS system with cooperative decision algorithm is presented. Each mobile host has to have IDS client installed, that runs a local detection engine that analyzes local data for anomalies. A cooperative detection mechanism decides whether there is intrusion detection, with all the nodes taking part in the decision process by voting. Anomaly detection model is used, as the authors argue that it is inefficient and insecure to rely on a database of attacks, due to a wide variety of wireless devices that make up an ad hoc wireless network. However, anomaly detection proves to result in poor performance and high false alarm rate. Another problem is monolithic IDS design. All the nodes have to accommodate IDS clients and take parts in global intrusion detection process. Clients are structured around several layers – MAC protocols, applications, system services, network monitoring, etc., and are self-contained monolithic entities, subject to attacks themselves.

To avoid problems outlined above, our approach would be to build a modular IDS system, based on intelligent mobile agents. Several IDS systems have been proposed that utilize mobile agents for wired networks [5, 6, 7, 8]. The main advantages of having a modular approach are increased fault tolerance, communications cost reduction, improved performance of the entire network, and scalability. As a voting system, some researchers used effective neural network classifiers [5], which were shown to minimize the number of false positive alerts while maintaining high intrusion detection rate.

3. Agent-based IDS for ad hoc wireless networks

This section introduces a proposed multi-sensor intrusion detection system employing cooperative detection algorithm. A mobile agent implementation is chosen, to support such features of the IDS system as mobility of sensors, intelligent routing of intrusion data throughout the network and lightweight implementation.

3.1. Modular IDS architecture

The proposed Intrusion Detection System (IDS) is built on a mobile agent framework. It is a non-monolithic system and employs several sensor types that perform specific certain functions, such as:

- Network monitoring: Only certain nodes will have sensor agents for network packet monitoring, since we are interested in preserving total computational power and battery power of mobile hosts.
- Host monitoring: Every node on the mobile ad hoc network will be monitored internally by a host-monitoring agent. This includes monitoring system-level and application-level activities.
- Decision-making: Every node will decide on the intrusion threat level on a host-level basis. Certain nodes will collect intrusion information and make collective decisions about network-level intrusions.
- Action: Every node will have an action module that is responsible for resolving intrusion situation on a host (such as locking-out a node, killing a process, etc).

Each module represents a lightweight mobile agent with certain functionality, making a total network load smaller by separating the functional tasks into categories and dedicating an agent to a specific purpose. This way, the workload of a proposed IDS system is distributed among the nodes to minimize the power consumption and IDS-related processing time by all nodes. A hierarchy of agents has been devised in order to achieve the above goals. Hierarchical IDS systems have been proposed in [5, 6, 7]. However, we will adapt our own hierarchy for our purposes. There are three major agent classes – monitoring, decision-making and action agents. Some are present on all mobile hosts, while others are distributed to only a select group of nodes, as discussed further. Monitoring agent class consists of packet, user, and system monitoring agents. The following diagram shows the hierarchy of agent classes.

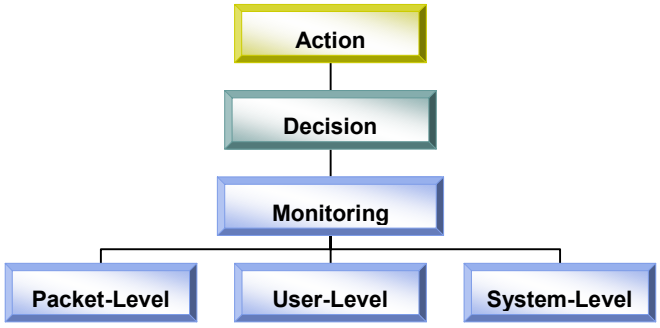

Figure 1. Layered Mobile Agent Architecture.

3.2. Agent distribution

As mentioned above, not all the nodes on a wireless ad-hoc network will host all types of IDS agents. To save the resources, some of the functionality must be distributed efficiently to a (small) number of nodes while

providing an adequate degree of intrusion detection. While all the nodes accommodate host-based monitoring sensors of an IDS, we use a distributed algorithm to assign only a few nodes to host sensors that monitor network packets, and agents that make decisions.

The idea is to logically divide a mobile network into clusters (similar to Clustered Gateway Switch Routing protocol described in [9, 10, 11] and used in [12] for authentication purposes) with a single cluster head for each cluster, and to monitor the packets within the cluster by only one node. The algorithm is presented below, along with an example.

Clustered Network-Monitoring Node Selection Algorithm

1. Hop Selection Step: based on security requirements, a certain number is selected as a number of hops. This step is important in choosing decision agent-hosting nodes, as well as network monitoring nodes, as selected number is the maximum number of hops from any node in the ad-hoc network to the Decision Node. Selection of this number greatly affects the network monitoring range, as only those nodes taking part in a decision process host network monitoring agents, resulting in lesser area of the network being monitored.

2. Let C_i denote the number of established connections (reachable nodes) for node i at the time of cluster setup, with a total of N nodes in the entire network. Each node sends its C_i value to all its reachable neighbors.

3. Upon receiving C_j values from its neighbors j, where $j \neq i$ for all $i = 1...N$, a node i sums up the total as S_i (connectivity index), which upon completion is broadcast to all nodes with a time to live (TTL) equal the number of hops selected in step (1):

$$S_i = C_i + \sum_j C_j \quad (1)$$

As an example, consider a connection graph of 11 nodes given in Figure 4. The number next to the node represents the connectivity index of that node.

4. Each node then has to vote to select cluster head node, that will accommodate network monitoring and decision agents. Every node sends a vote packet to the node it selects based on highest connectivity index received as a result of a broadcast in step (3). If a node receives a vote from a node with equal S_i value, it doesn't send a vote to the source node. In case two nodes have equal S_i values and send votes to each other simultaneously, the node with the largest total of S_i values sends a "discard vote" message to the other node. This will ensure that the minimal number of nodes is selected for hosting packet-monitoring agents. Note that in step 3, a node will decrease TTL count and broadcast the packet containing S_i to all it's reachable neighbors, resulting in every node receiving the information about the maximum S_i within the hop distance.

5. Each node that received at least one vote, loads and runs Network Monitoring and Decision Agents. Steps (4) and (5) are shown on a diagram below, giving scenarios for (a) one-hop and (b) two-hop ad-hoc wireless networks.

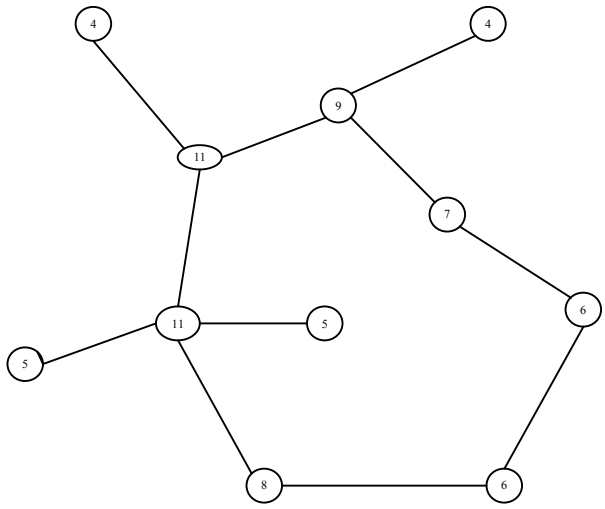

Figure 2. Computing Connectivity Index.

The selected nodes host network-monitoring sensors that collect all packets within communication range, and analyze them for known patterns of attacks. Parameters such as per-protocol statistics, number and frequency of certain packet types and consistency with the model are verified.

The main advantage of the allocation algorithm above is that overall packet-monitoring task is limited to a small subset of nodes, thus conserving power and processing capabilities for many nodes in the ad hoc network.

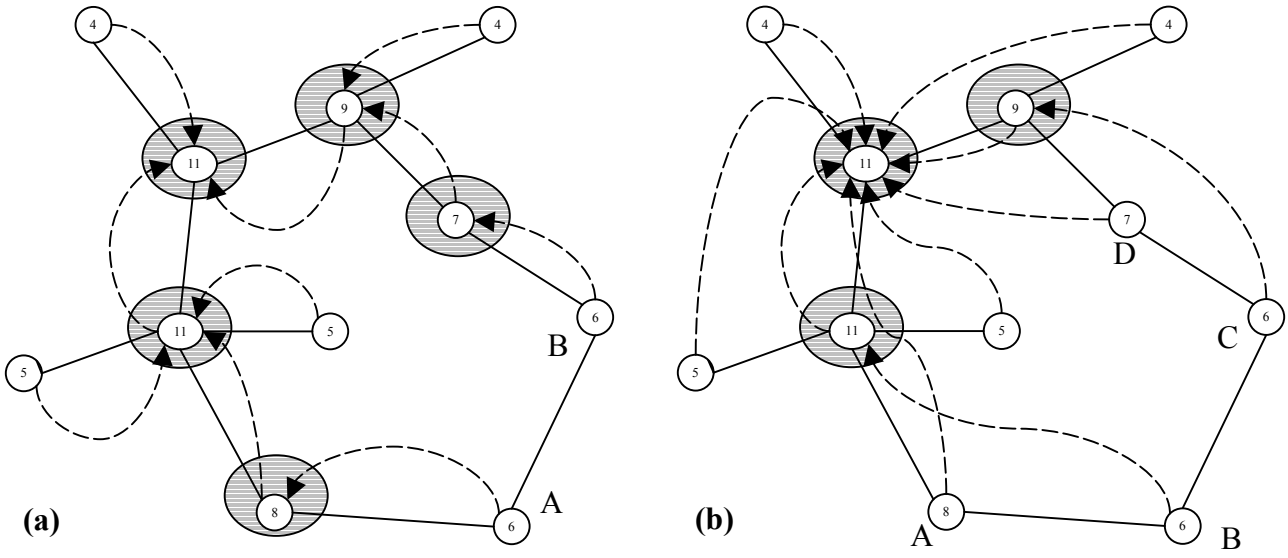

Figure 3. Network monitoring node selection with (a) one-hop radius, and (b) two-hop radius. Dashed lines indicate a vote packet route. Nodes selected to host network monitoring and decision agents are highlighted.

As the physical network arrangement changes, cluster membership is dynamically updated. The figure 4 below shows a percentage of nodes engaged in network-monitoring activities vs. the total number of the nodes.

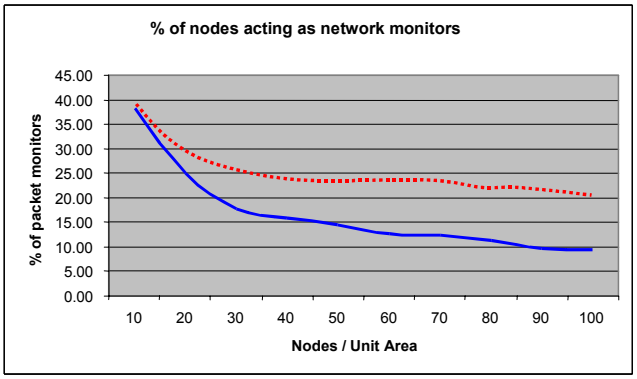

Figure 4. Percentage of nodes engaged in packet monitoring in a one-hop (dashed line) and two-hop (solid line) network.

3.3. Activity-monitoring process

As shown in Figure 1, monitoring agents are categorized into packet monitoring sensors, user activity sensors and system-level sensors. While packet monitoring is activated only when a node participates in the network (is a member of a cluster), local activity sensors are present on each node and are active all the time. Each sensor performs certain level of monitoring activity and reports anomalies to the decision agents.

Packet-monitoring agents reside on each selected node. On the Figure 3 above, we can see that for a case of one-hop cluster, 5 nodes out of a total of 11 nodes host network monitoring sensors, resulting in the entire network being monitored. For instance, a packet sent from node A to node B will be received and analyzed by the monitoring node to the left of node A. In fact, for a case of one-hop cluster, every node has at least one neighboring node hosting a packet monitoring agent, and thus the entire network is always being monitored. If the system resources are scarce and security requirements can be relaxed, a two-hop system will be more appropriate, as indicated on Figure 3(b). Here, we have only 3 hosts dedicated to packet monitoring and decision-making process, saving overall system resources. However, in this scenario, 3 links are not being monitored, which may be acceptable for a highly-dynamic environment, where network configuration changes often. The rationale is that a node is located in close proximity (within two hops) to the packet-monitoring node, and rapid movement may position the node within a communication range of that packet-monitoring node.

Each cluster head monitors packets sent by every member of its cluster, and therefore, the agent subsystem has a low-level access to the underlying operating system's network layer to capture packets that are not

intended for the cluster head node. For now, we limit the collection of packets only to those that have as originator any node that belongs to the cluster. This is done to prevent processing of the same packet more than once by any packet-monitoring agent. When packets are captured, they are inserted in a queue (logically), and physically added to a buffer of fixed size (the size depends on the node's available memory). The packets are then dequeued and processed by the agent's case-based reasoning engine for intrusion detection. If a queue becomes full, further packets are dropped until space is available in the queue (see Figure 5). By varying queue size, we limit processing done by a cluster head node, as its resources are also used for performing regular user tasks. Agent subsystem also allows us to limit CPU usage by an agent to a certain level, acceptable by the user.

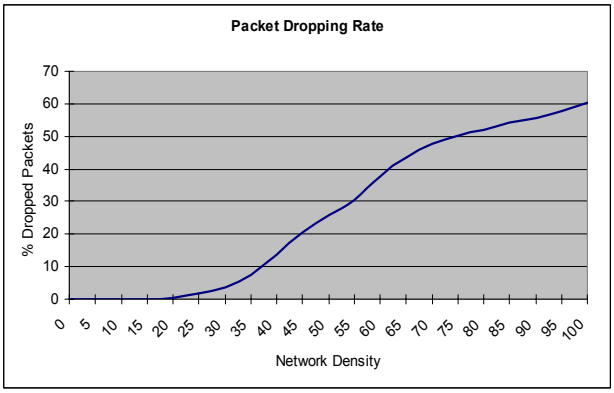

Figure 5. Increase in packet dropping rate as the network density increases.

Local detection agents are located on each node of an ad-hoc network, and act as user-level and system-level anomaly-based monitoring sensors. These agents look for suspicious activities on the host node, such as unusual process memory allocations, CPU activity, I/O activity, user operations (invalid login attempts with a certain pattern, super-user actions, etc). If an anomaly is detected with strong evidence, a local detection agent will terminate suspicious process or lock out a user and initiate re-issue of security keys for the entire network. If some inconclusive anomalous activity is detected on a host node by a monitoring agent, the node is reported to the decision agent of the same cluster that the suspicious node is a member of. If more-conclusive evidence is gathered about this node from any source (including packet monitoring results from a network-monitoring agent), the action is undertaken by the agent on that node, as described above.

4. Intrusion Detection

4.1. Collaborative vs. independent decision making

Experience with intrusion detection systems designed for wired networks helps us to classify decision-making mechanisms for such IDS systems into two categories – collaborative and independent. The first type of decision-making mechanism is employed in a system where each node can take active part in intrusion detection process. An example of such a system is given in [3], where a simple majority voting scheme is used, in which any node that detects an intrusion with high enough confidence can initiate a response. More sophisticated cooperative decision-making schemes use fuzzy logic and rules to determine the threat level more accurately and initiate intrusion response. Such mechanisms are discussed in [4] and [5]. However, such systems are prone to denial of service and spoofed intrusion attacks, where any (malicious) node can trigger full-forced intrusion response, affecting entire network. In an independent decision-making system, certain nodes are designated to perform decision-making functionality. Their task is to obtain intrusion alert information from other nodes and to decide with a good accuracy whether or not a node in question presents a threat to network security. Other nodes don't have any influence on the decision-making process that concerns a certain node. This category of decision-making mechanisms is far less prone to spoofing attacks; however, the amount of information obtained by a decision-making node about each node participating in the network is limited. If a node in question had failed in local intrusion detection and all reporting mechanisms were somehow disabled, it will be difficult to detect such kinds of passive intrusion, where, for instance, a node could be intruded in and used as a passive listener on the network.

4.2. Intrusion detection process

Our intrusion detection system utilizes a customized independent decision-making mechanism. Decision agents are located on the same nodes as packet-monitoring agents. Detection and classification of security violations works as follows. Decision agent contains a state machine for all the nodes within the cluster it resides in. As intrusion or anomalous activity evidence is gathered for each node, the agent can decide with a certain confidence that a node has been compromised by looking at reports from the node's own local monitoring agents, and the packet-monitoring information pertaining to that node. There is no need for other neighboring nodes to detect an intrusion or anomalies from the node in question, as this will be subject to denial of service (DOS)

attacks on such a decision scheme. When a certain level of threat is reached for a node in question, decision agent dispatches a command that an action must be undertaken by the local agents on that node, as described in section 3. In time, the threat level decreases for each node in the decision agent's database. This is necessary to account for certain uses of the network node that do not conform to accepted range of normal behavior, yet do not represent a threat to the wireless network as such.

Local anomaly detection models have been developed [3, 4, 5] that can detect an intrusion with a great degree of accuracy. According to the surveyed research, two types of profiling are made. Some IDS systems maintain a database of possible intrusion activity patterns and trigger alarm when such activity is detected. These systems result in fewer false alarms due to a variation in node usage patterns; however, intrusion activities with new patterns are likely to be underreported. The other category of IDS systems maintain a normal operational profile formed by a learning process. Anything that falls outside such a profile of activities is classified as a possible intrusion. These systems have a higher false alarm rate, but are more likely to discover unknown intrusion, making such a model a choice for our IDS.

5. Conclusions

With emergence of a wide range of wireless devices, protecting ad-hoc wireless networks became an increasingly important but also a more difficult task. Scarce computational and power resources of mobile nodes impose heavy limitations on functionality of an effective intrusion detection system. Given these limitations, we have proposed a distributed modular IDS system designed for ad hoc wireless networks. This architecture is aimed to minimize the costs of network monitoring and maintaining a monolithic IDS system, also providing a degree of protection against the intruder. New agents with added functionality can be plugged in when an expansion is necessary. Moreover, based on individual security requirements, the level of monitoring can be decreased resulting in greater availability of computational resources for the entire network.

Future work will involve research into more robust and intelligent cooperative detection algorithms, as well as a choice of an anomaly detection model most appropriate for our IDS system. Some of the work has been presented in [3, 4, 5].

Whereas an IDS system may detect attacks on mobile hosts, another possibility is to attack IDS system itself. As our system employs mobile agents for intrusion detection, these mobile agents may be the primary target of an attack [13, 14]. We will investigate possible attacks on our IDS system infrastructure and on individual mobile agents in particular, and research effective means of defense.

Acknowledgements

This work was supported by the US Army Research Office, grant number DAAD19-01-1-0502. The views and conclusions herein are those of the authors and do not represent the official policies of the funding agency.

References

[1] R. Lippmann et. al., "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation", Proceedings of DARPA Information Survivability Conference & Exposition II, Volume: 2, 1999, pp. 12-26.

[2] J. Haines, L. Rossey, R. Lippmann, R. Cunningham, "Extending the DARPA Off-Line Intrusion Detection Evaluations", Proceedings of DARPA Information Survivability Conference & Exposition II, Volume: 1, 2001, pp. 35-45.

[3] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks", Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom'2000, pp. 275-283.

[4] A. Siraj, S. Bridges, R. Vaughn, "Fuzzy Intrusion Detection", Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Volume: 4, 2001, pp. 2165-2170.

[5] D. Dasgupta and H. Brian, "Mobile Security Agents for Network Traffic Analysis", Proceedings of DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01, Volume: 2, 2001, pp. 332–340.

[6] M.C. Bernardes and E. Santos Moreira, "Implementation of an Intrusion Detection System based on Mobile Agents", Proceedings of International Symposium on Software Engineering for Parallel and Distributed Systems, 2000, pp. 158-164.

[7] G. Helmer, J. Wong, V. Honavar, L, Miller, "Lightweight Agents for Intrusion Detection", Technical Report, Dept. of Computer Science, Iowa State University, 2000.

[8] J. Tao, L. Ji-ren, Q. Yang, "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, 2000, pp. 134-139.

[9] E. Royer, C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communications, Volume: 6 Issue: 2, April 1999, pp. 46-55.

[10] C.-C. Chiang, et. al., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", Proceedings of IEEE SICON, April 1997, pp. 197-211.

[11] R. Ramanujan, A. Ahamad, J. Bonney, R. Hagelstrom, K. Thurber, "Techniques for Intrusion-Resistant Ad Hoc

Routing Algorithms (TIARA)", Proceedings of 21st Century Military Communications Conference, Volume: 2, 2000, pp. 660-664.

[12] L. Venkatraman, D. Agrawal, "A Novel Authentication Scheme for Ad Hoc Networks", Proceedings of Wireless Communications and Networking Conference, Volume: 3, 2000, pp. 1268-1273.

[13] X. Guan, Y. Yang, J. You, "POM-A Mobile Agent Security Model against Malicious Hosts", Proceedings of the 4th International Conference on High Performance Computing in the Asia-Pacific Region, Volume: 2, 2000, pp. 1165-1166.

[14] M. Chun Man, V. K. Wei, "A Taxonomy for Attacks on Mobile Agent", Proceedings of International Conference on Trends in Communications, Volume: 2, 2001, pp. 385-388.