

Case-Based Agents for Packet-Level Intrusion Detection in Ad Hoc Networks

R. Guha and O. Kachirski
SEECs - Computer Science
University of Central Florida
Orlando, FL, USA
Contact: guha@cs.ucf.edu

D.G. Schwartz, S. Stoecklin, and E. Yilmaz
Department of Computer Science
Florida State University
Tallahassee, FL, USA
Contact: schwartz@cs.fsu.edu

Abstract

In this paper we propose a distributed agent framework for an intrusion detection system aimed at ad hoc wireless networks. Wireless networks are particularly vulnerable to intrusion, as they operate in an open medium, and use cooperative strategies for network communications. By efficiently merging audit data from multiple security agents, we analyze the entire ad hoc wireless network for intrusions and try to inhibit intrusion attempts. A case-based reasoning approach to our intrusion detection engine provides a framework incorporating sophisticated artificial intelligence techniques that help overcome some of the limitations of other rule-based intrusion detection systems. In contrast to many existing intrusion detection systems, we design and implement an efficient, bandwidth-conscious framework that targets intrusion at multiple levels and takes into account the distributed nature of ad hoc wireless network management and decision policies.

1. Introduction

1.1. Case-Based Reasoning

The basic features of a general case-based reasoning (CBR) system are depicted in Figure 1. The most important component of the system is the case archive where the previously experienced problems are stored with their solutions. Each entry in the case archive is called a “case” which contains (i) the features describing the problem, and (ii) the action or actions that were taken to solve the problem. When a problem is detected in the surrounding environment, it is formulated as a set of case features (step 1.0). Then, this problem description is transferred to a search engine that extracts the similar cases from the case archive, where similarity is measured by the similarity between the matching features of the problem description and the case features of actual cases

in the case archive (step 2.0). The returned cases are ranked according to their degrees of similarity to the given problem. At this moment two different scenarios are possible: either some of the selected cases are decided as a solution to the problem or a new case is formulated to solve the problem based on the returned cases. In either case, the actions recommended by the returned case or cases are taken (step 4.0). Furthermore, the measure of success or failure of the result of the action is reported along with the case into the case archive (step 5.0). In future applications of the case-based reasoner, this information will be taken into account in the similar-case extraction process so that the performance of the system will improve over time.

1.2. Intrusion Detection Systems

Traditionally, intrusion detection systems (IDSs) were divided into two classes: network-based and host-based IDS. Network-based systems (NIDS) listen on the network, and capture and examine individual packets flowing through a network. NIDSs often require dedicated hosts and special equipment, and can be prone to the network attack. A few reliable NIDSs are described in [1, 5, 7, 8]. 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. These systems can considerably slow down the hosts that have IDS clients installed.

IDS systems are functionally divided into two categories: anomaly detection and misuse detection systems. Anomaly detection bases its ideas on statistical behavior modeling. This model detects intrusion detections in a very accurate and consistent way, and has a low number of false alarms if the system under

surveillance follows static behavioral patterns. 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. However, a frequently updated (and large) database of known attack signatures must be maintained. Both categories of IDS can be used on host-based and network-based IDS systems.

Structural and behavioral differences between wired and wireless mobile networks make existing IDS designs inapplicable to wireless networks. As discussed above, wireless network communications are conducted in an open air environment. Thus, network monitoring in wireless ad hoc networks is performed at every network node [3]. This approach is inefficient due to network bandwidth consumption and increased computations – resources that are highly limited in a wireless network. Host-based monitoring also contributes to a high amount of processing on each host, shortening battery life and slowing down the host. Physical mobile host security is an issue, as each host contains keys used to encrypt

information over the network, and if captured, the network is subject to eavesdropping.

Applying functionality-based network IDS models also has limitations. The anomaly detection model is built on a long-term monitoring and classifying of what is a normal 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 an ad hoc network would have to be replicated among all the hosts.

To avoid the problems outlined above, we have designed a modular IDS system [16], based on intelligent mobile agents. We have also developed a case based approach to network intrusion detection [15]. In this paper we incorporate the case-based reasoning engine, proposed in [15] for detecting intrusions at the packet level in the modular IDS system [16] and study the effect of such incorporation.

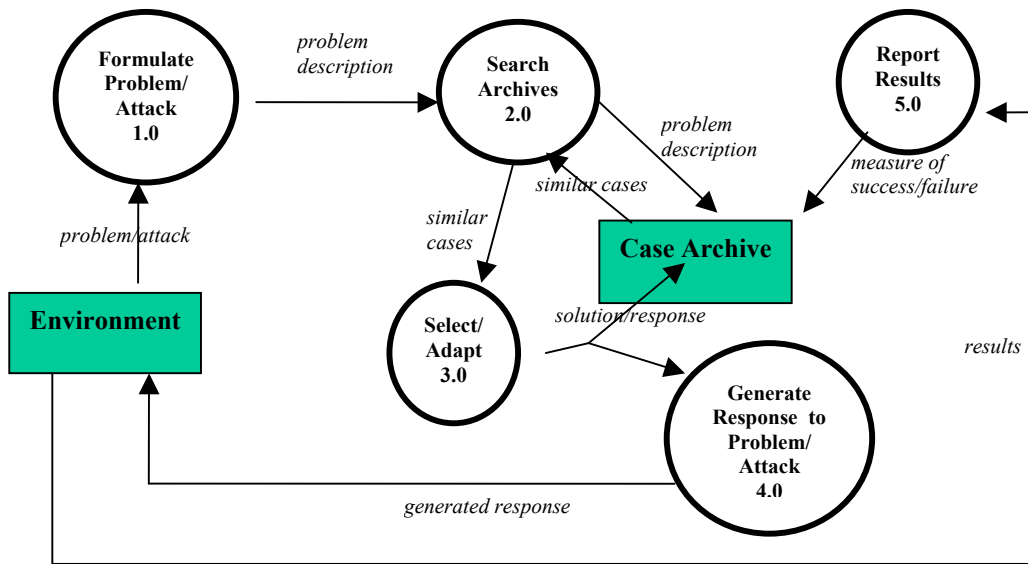


Figure 1. Case-Based Reasoning Process.

2. Case-Based Agent Intrusion Detection System

2.1. Agent-Based IDS architecture

Our IDS is built on a mobile agent framework, as described in [16]. It is a non-monolithic system and employs several sensor types that perform specific functions, such as:

- Network monitoring: Only certain nodes 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 is monitored internally by a host-monitoring agent. This includes monitoring system-level and application-level activities.
- Decision-making: Each node determines on its own degree of intrusion threat. Certain nodes collect

intrusion information and make collective decisions about network-level intrusions.

- Action: Every node has an action module responsible for resolving intrusion situations on a host.

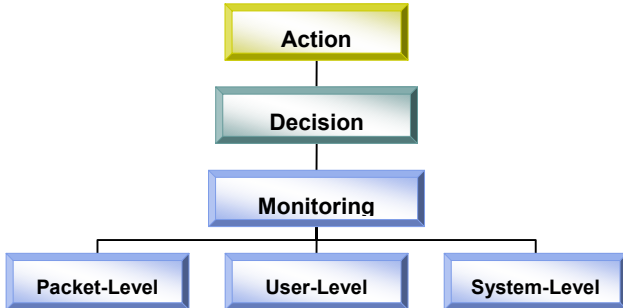


Figure 2. Layered Mobile Agent Architecture.

An advantage of the agent-oriented approach is that one can make the total network load smaller by separating the necessary functional tasks into categories and dedicating different agents to different specific purposes. This way, the workload of the IDS system is distributed among the nodes in such a way as 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. This is depicted in Figure 2. There are three major agent categories: 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. The monitoring agent class consists of packet, user, and system-monitoring agents. Hierarchical IDS systems have been proposed in [5, 6, 7].

To save resources, some of the IDS 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 the IDS, we use a distributed algorithm described in [16] to assign a few nodes to serve as host sensors that monitor network packets, and agents that make decisions. We logically divide a mobile network into clusters (similar to the Clustered Gateway Switch Routing protocol described in [9, 10, 11, 12]) with a single cluster head for each cluster that monitors packets within the cluster. This will ensure that the minimal number of nodes is selected for hosting packet-monitoring agents.

Packet-monitoring agents reside on each selected cluster head, collect all packets within communication range, and analyze them for known attacks. As the physical network arrangement changes, cluster membership is dynamically updated. For the case of a one-hop cluster, each 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 is more appropriate. At any given time, a few links might not be monitored. This may be acceptable for a highly-dynamic environment, where network configuration changes often. The packet detection mechanism is discussed further in this paper.

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, and 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 processes 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 cluster of which the suspicious node is a member. 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 action agent on that node, as described above.

2.2. Case-Based Detection Mechanism

2.2.1. Generic Case Based Reasoning

Case-based reasoning systems are designed for a given application domain. But it is possible to abstract out the common aspects of CBR from the domain specific aspects [15]. This leads to a generic case-based reasoner from which any arbitrary domain-specific case-based reasoner can be created as a specific instance.

In order to build the desired generic case-based reasoner, it is required to generalize both the notion of a case and the notion of a similarity metric used for determining the degree of similarity between cases. Since cases are described by their features, the first task is to describe the generic notion of a case feature. The framework proposed in [15] makes it possible to define virtually any type of feature and any type of case.

The similarity metric for cases can be defined as a collection of feature comparison results with a rule specifying how these intermediate results are combined. Moreover, feature comparisons can be generalized into the generic notion of feature comparator of which each specific comparator is an instance. Although each case feature may require a different type of comparison, the result of the comparison should be a similarity assessment between the same case feature of the problem specification and of a case from the case archive.

Therefore for each new type of case feature, it is required to define a comparator that determines the degree of similarity between the problem situation feature and the corresponding feature in the case archive. Since different case features may use the same comparator, the number of comparators in the system will be much smaller than the number of different features. The modular distinction between comparators and case features simplifies the adaptation of the system into different problem domains.

2.2.2. Adaptive Case-Based Reasoning Process

The generic CBR component assumes no knowledge about the application domain regarding case features and their comparison. The system can be tailored into a domain-specific case-based reasoner by defining the data type definition of the XML representation of a domain specific case, together with a metadata dictionary where the data about different case features, such as the required comparator and its value type, are stored. This use of a metadata dictionary and the separation of domain specific knowledge from generic components is an example of “adaptive” or “reflective” architecture. Hence the title for our system The advantage of this software engineering approach is that the same generic CBR source code can be used for any application domain – no, or very minimal, additional programming is required.

The adaptive CBR process of an IDS is as follows. A packet is received from the network and fed into the CBR module. The packet is converted into an XML representation as specified in a corresponding DTD file. The search engine in the CBR module searches for similar cases in case archive. In the search process, cases are compared to the received packet’s XML representation. For each feature in the packet XML data, the CBR module looks in the metadata dictionary for the type of the required comparator, and the comparator is created by reflection (i.e., during run time). Each comparator determines whether the packet feature matches the corresponding case feature. Once all the features are compared, the CBR module assigns a similarity value for the compared case. Last the CBR module retrieves the matching case or cases and performs the prescribed action.

3. Packet-Level Intrusion Detection

3.1. CBR Implementation of a Packet-Monitoring Agent

In order to implement a packet-monitoring agent using adaptive CBR, we converted the rules of the well-known Snort IDS into a case archive. All the elements in a rule header, as well as the rule options, that are used in

the rule matching process by Snort are treated as case features, and the rule action and its corresponding rule options (such as message element) are treated as the case action.

In the Snort IDS, the corresponding rule action is taken only if all of the elements that make up a rule match with the network packet. This means that in the corresponding domain-specific CBR system, the similarity metric must be bivalent; in other words, the matches must be exact. Hence there is no need for similarity ranking. Thus, from a CBR standpoint, the corresponding system is quite simple.

The entire packet monitoring CBR system, including the case archive created from Snort rules, together with required comparators, is quite small in size. Moreover, due to its modular design and implementation, both the reasoner and case archive components are portable. This is very important for network agent implementation. As explained in the next section, this CBR system can serve as the core of the “decision agents” in their intrusion detection process.

3.2. Packet Monitoring and Evidence Collection Process

Packet-monitoring agents reside on cluster head nodes. 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 3). 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. As case-based reasoner scans the packets from the queue, it checks the packet information (network addresses, ports and payload contents) against a known set of rules, as described in the previous section. When a match is detected, alert is raised and forwarded to the decision-making agent (residing on the same host – the cluster head).

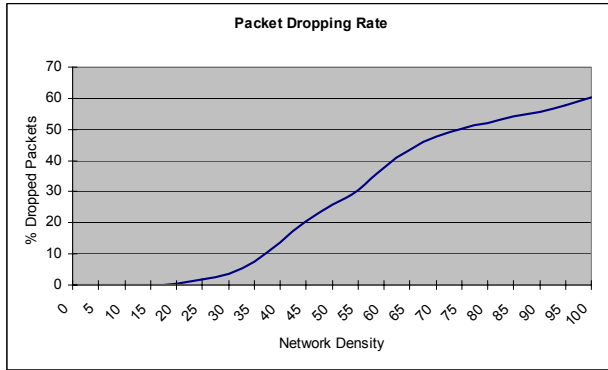


Figure 3. Increase in packet dropping rate as the network density increases.

Our intrusion detection system utilizes an independent decision-making mechanism. Decision agents are located on the same nodes as packet-monitoring agents. Decision agent contains a state machine for all the nodes within the cluster it resides in. As intrusion and 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. 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. Decision-making agent maintains a "sliding-window" view on the intrusion data for each node within its cluster. 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. Repetitive alerts of the same type within that window will cause an action to be undertaken by a decision agent to secure the breach in a network caused by a certain node or a group of nodes.

4. Concluding Remarks

In this paper, we have proposed a distributed modular intrusion detection system that employs case-based reasoning engine for intrusion detection. The database for CBR engine is distributed to every node of the wireless ad hoc network. The CBR engine is activated by a packet monitoring agent of a cluster-head node. Using our network clustering algorithm, developed in [16], we have simulated the intrusion detection process that uses CBR engine and an intrusion database. The simulation shows that for a single cluster head node processing all the incoming packets, the number of dropped packets increases significantly when the density

of ad hoc wireless network increases. Although each cluster head's processing load due to IDS system is controlled, the reliability of IDS system attack detection decreases. This issue will be addressed in future work, when cluster processing algorithm will be employed, where the monitoring tasks are shared among nodes in a cluster. Current and future work involves investigating the most effective sharing process, resulting in a minimized packet drop rate and providing a high degree of protection, while limiting the load on each node due to intrusion detection processing.

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] Lippmann R., 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] Haines, J., L. Rossey, R. Lippmann, and 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] Zhang, Y. 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] Siraj, A., S. Bridges, and R. Vaughn, "Fuzzy Intrusion Detection", *Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Volume 4*, 2001, pp. 2165-2170.
- [5] Dasgupta, D. and H. Brian, "Mobile Security Agents for Network Traffic Analysis", *Proceedings of DARPA Information Survivability Conference & Exposition II, DISCEX '01, Volume: 2*, 2001, pp. 332-340.
- [6] Bernardes, M.C. 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] Helmer, G., J. Wong, V. Honavar, L. Miller, "Lightweight Agents for Intrusion Detection", Technical Report, Dept. of Computer Science, Iowa State University, 2000.

- [8] Tao, J., L. Ji-ren, and 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] Royer, E. and C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", *IEEE Personal Communications*, Vol. 6, No. 2, April 1999, pp. 46-55.
- [10] Chiang, C.-C., et. al., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", *Proceedings of IEEE SICON*, April 1997, pp. 197-211.
- [11] Ramanujan, R., A. Ahamad, J. Bonney, R. Hagelstrom, and 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] Venkatraman and L., D. Agrawal, "A Novel Authentication Scheme for Ad Hoc Networks", *Proceedings of Wireless Communications and Networking Conference, Volume 3*, 2000, pp. 1268-1273.
- [13] Guan, X., Y. Yang, and 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] Chun Man, M. and V. K. Wei, "A Taxonomy for Attacks on Mobile Agent", *Proceedings of International Conference on Trends in Communications, Volume 2*, 2001, pp. 385-388.
- [15] Schwartz, D.G., Stoecklin, S., and Yilmaz, E., "A Case-Based Approach to Network Intrusion Detection", *Fifth International Conference on Information Fusion, IF'02*, Annapolis, MD, July 7-11, 2002, pp. 1084--1089.
- [16] Kachirski, O. and Guha, R., "Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks", *Proceedings of IEEE Knowledge Media Networking Conference, KMN'02*, July 2002