A Mechanism for Detecting and Responding to Misbehaving Nodes in Wireless Networks

Damon McCoy, Doug Sicker, Dirk Grunwald
Department of Computer Science
University of Colorado
Boulder, Colorado 80309-0430
Email: {damon.mccoy,sicker,grunwald}@cs.colorado.edu

***Abstract*—While mechanisms exist to instantiate common security functionality such as confidentiality and integrity, little has been done to define a mechanism for identification and remediation of devices engaging in behavior deemed inappropriate. This ability is particularly relevant as devices become increasingly adaptive through the development of software-defined and frequency agile radios. Adaptive devices can alter their behavior in a way that is noncompliant to a given set of standards and thus cause problems for other compliant devices.**

We address this deficiency by developing and assessing a mechanism for detecting misbehaving nodes in wireless systems. While we developed our system on an 802.11 network, the same approach could readily be applied to other wireless networks. Our mechanism is based on a reputation-enabled intrusion detection system, in which a centralized trust authority monitors traffic and collects secondhand information on potentially misbehaving nodes. The system integrates a mixture of alarms and reports to calculate a reputation vector of all nodes in the system. An XML based policy engine is used to detect policy violations. These mechanisms are built to be flexible and extensible in order to deal with the issues arising out of software programmable devices. In extending beyond traditional intrusion detection, our approach will incorporate physical layer information, such as power and frequency use, in determining improper behavior. In evaluating the system, we consider how our mechanism, 1) impacts system performance, 2) correctly identifies misbehaving nodes, 3) addresses ”bad mouthing” and 4) resists collusion.

I. Introduction

Wireless networks are emerging as a common method to enable and extend communication in mission critical communications infrastructure. They are commonly used in such broadly ranging and important areas as health care, public safety and national defense. While these wireless networks are becoming more pervasive, they are also undergoing a radical change in how they operate. Traditional wireless devices operate in fixed Radio Frequency (RF) bands and follow fairly rigid network protocols. However, emerging wireless devices are becoming increasingly agile and will soon support operation across multiple RF bands. [1] Such adaptations could readily lead to devices operating in a way that violates expected protocol behavior.

While many networks implement strong mechanisms for confidentiality, integrity, and authentication, some of the most disconcerting issues arise from authenticated and trusted devices that are misconfigured or fall into the hands of a misbehaving or malicious user. Examples of misconfigured devices include transmitting at a higher than legal power level or transmitting on the wrong frequency (possibly due to variation in spectrum policies across countries). Misbehaving devices can send messages that violate the network protocol, provide incorrect information, or launch a Denial of Service (DoS) attack. For example, the WiFiHog [2] is a device designed to gain complete control over a public access wireless network. A combined reputation and behavior detection system is needed to detect and remediate devices that violate the policies of the network.

In this paper, we describe a Misbehaving Node Detection (MIND) mechanism, which integrates policy, detection, and remediation components for identifying and handling misconfigured, misbehaving, or malicious devices. In designing this system, we identified five key issues our system should address.

1) The system should provide a policy based mechanism for identifying misconfigured, misbehaving, and malicious nodes.
2) The system should be able to detect and remediate nodes that violate policy.
3) The system should be flexible and sufficiently extensible to handle common attacks and misconfigurations in wireless networks.
4) The system should not burden the device or the network in terms of additional computation, storage, and message complexity.
5) The system should be robust to malicious individuals and collectives of nodes that attempt to subvert the system.

To ease our development efforts, we made use of an 802.11 system; however, our approach could readily be applied to other single-hop wireless networks. MIND is suited to smaller infrastructure and single-hop ad-hoc networks where all members are initially trusted and can be authenticated using public key cryptography [3], and messages are protected by strong confidentiality, integrity and availability mechanisms. The system is assumed to have one fully trusted centralized node that monitors wireless traffic using wireless snort [4], an open source wireless Intrusion Detection System (IDS), and also aggregates secondhand reports from nodes in the network concerning other nodes that are misbehaving or acting maliciously.

An XML based policy engine is used to detect policy violations including DoS and active attacks against the network. Detection policies are transformed into wireless snort rules to enable detection of policy violations. The centralized authority integrates a mixture of alarms generated by the wireless snort system and reports from authenticated devices to create a global reputation vector. This vector is distributed by the centralized authority to all members of the network. Authenticated nodes use a fusion of the global trust vector and their local trust vector to decide what level of trust to assign to a device. It is a hybrid approach of centralized and distributed local trust that protects against a number of attacks. The policy engine also includes rules to specify what actions, if any, should be taken by the remediation system when bad actors are detected. Our system differs from general intrusion detection by incorporating physical layer detection information (such as frequency and power) with traditional upper layer detection information. To our knowledge, this is the first implementation of a wireless reputation system that detects and remediates misbehaving and malicious nodes. We evaluated our system against a mixture of misconfigured and malicious nodes, a bad mouthing attack, and a Distibuted Denial of Sevice (DDoS) attack. Our evaluations of the system show that in all three experiments, the MIND system correctly identified the malicious nodes and removed them from the system after other failed attempts at remediation.

The rest of this paper is organized as follows. Section II provides background on the existing systems and techniques incorporated into MIND. Section III lays out the design and implementation details of the system. In section IV, the system is evaluated to measure its performance overhead and successful detection of bad actors. Section V presents related work and we conclude the paper in section VI.

II. Background

Parts of the MIND system are based on and adapted from a number of preexisting standards, systems, and algorithms. This section includes background information on intrusion detection systems, reputation systems, and the use of XML for defining policies.

A. Intrusion Detection Systems

Snort [5] is an open source Intrusion Detection System (IDS) that uses a flexible rule based detection architecture to identify intrusions in networks. The core Snort IDS currently supports wired networks exclusively. Wireless Snort [4] is an extension for the Snort IDS that adds support for IEEE 802.11 wireless frames. Wireless Snort allows for 802.11 specific detection rules through a new ”WiFi” rule protocol. As we demonstrate in Section 3, the extended rule set proved to be sufficiently flexible to allow for a combination of automated and manual translations of policies written in XML into rules for detecting policy violations.

B. Reputation Systems

A large portion of previous work in reputation systems has focused on the area of peer-to-peer [6], [7], [8], [9], multi-hop wireless networks [10], [11], [12], [13], or more general work in the area of evidence [14] and reputation [15]. The Eigenvector trust algorithm [16] was originally developed for reputation management in peer-to-peer networks to reduce the number of inauthentic files on the network. The algorithm uses a distributed method to compute global trust vectors for members of the network. The trust vectors are composed of firsthand and secondhand knowledge of nodes in the system. The analysis of the algorithm demonstrates that it is effective at detecting malicious nodes and resisted collectives of peers attempting to subvert the system. We make use of a variation of the EigenTrust method to compute trust vectors in the MIND system.

C. XML Defined Policies

The Extensible Markup Language (XML) is a standardized language for describing structured information in documents. It provides a means for defining tags (i.e., the labels that associate with content in a document) and the structural relationships among these tags. [17] Beyond its common use in facilitating the exchange of richly structured content on the web, XML is gaining popularity as a means for defining and exchanging structured policy information. The Platform for Privacy Preferences (P3P) makes use of an XML schema to describe privacy policies that are machine readable. [18] The OASIS Technical Committee produced an XML based language, the extensible Access Control markup language (XACML), for defining authorization polices for accessing resources.[19] Lastly, the DARPA neXt Generation (XG) Communications project is using OWL, a web ontology language, to define a policy structure for the operation of frequency agile radio. [20] OWL is similar to XML but provides richer semantics.

III. Design and Implementation

In this section, we begin by presenting an attack model and system assumptions. We then describe in detail the different components of the MIND system, which includes the XML policies, the detection mechanisms, the trust algorithms, and the remediation mechanisms.

A. Attack Model

We assume that all nodes in the system are initially trusted and authenticated using public key cryptography. Over time, nodes may continue to behave properly, or they may become misconfigured, misbehaving, or malicious. The misconfigured and misbehaving nodes may attempt to correct their behavior either on their own or when the centralized authority sends them a remediation notice. (Of course, some nodes may not be able to change their behavior because of hardware or software problems). Malicious nodes are likely to continue to violate the policy of the network even after receiving a remediation notice from the centralized authority. There can

```
<MessageLimit>
   <min> 30 </min>
   <messages> 400 </messages>
   <ReputationDecrease>0.2
       </ReputationDecrease>
</MessageLimit>
<InvadeFrame>
    <ReputationDecrease>0.05
        </ReputationDecrease>
</InvadeFrame>
<IncorrectInformation>
    <ReputationDecrease>0.1
        </ReputationDecrease>
</IncorrectInformation>
...
```

Fig. 1. Example XML detection rules

```
<Notify>
  <Threshold>0.8</Threshold>
</Notify>
<Reboot>
  <Threshold>0.6</Threshold>
</Reboot>
<RadioSilence>
  <Threshold>0.4</Threshold>
  <Duration>
      <hour>1</hour>
  </Duration>
</RadioSilence>
<Remove>
  <Threshold>0.2</Threshold>
</Remove>
```

Fig. 2. Example XML remediation rules

be individual misbehaving, misconfigured, or malicious nodes, or a collective of malicious nodes working in conjunction to subvert the system. It is assumed that there is a single centralized authority that is trusted at all times and does not misbehave in any way.

B. *System Assumptions*

We assume that all nodes have the public key of the centralized authority and the centralized authority has the public keys of all the nodes in the system. It is assumed that the system uses strong confidentiality, integrity, and availability algorithms to protect the system from non-member nodes. These assumptions are outside the scope of this paper, but have been addressed in the system design. It is also assumed that the network is a small infrastructure or single-hop ad-hoc wireless network.

C. *XML Policy*

A simple yet flexible set of XML tags are defined to specify detection and remediation rules for the MIND system. These tags include detection rules for DoS attacks, protocol violation, frame formatting errors, and messages containing incorrect information. The set of policy XML tags also contains tags for remediation of misbehaving or malicious nodes.

The detection rules define network policies and expected user behavior. All detection rules include a mechanism for decreasing a node's reputation due to policy violations. Some alarms (i.e., captured violation events) contain additional parameters. For example, the MessageLimit tag includes two additional tags that define how many messages in a fixed time period constitute a DoS attack. Example detection rules are shown in Figure 1.

In order for network policies to be meaningful there must be defined consequences for misbehavior. The remediation tags include a reputation threshold, whereby if a node's reputation falls below this threshold the remediation action is taken by the centralized authority. Example remediation rules are shown in Figure 2.

We show in our evaluation section that these policy rules are robust and capable of detecting and remediating a number of common attacks.

D. *Detection*

As shown in Figure 3, the centralized authority implements the bad behavior detection mechanism through the use of a snort engine that combines firsthand and secondhand information. The firsthand information is gathered by the centralized authority. The secondhand information is gathered by the aggregation system that also runs on the centralized authority and accepts reports of misbehavior from other nodes in the system. Both of these mechanisms are important and together can detect a large amount of the misbehavior within the network. The wireless Snort system, for instance, can detect DDoS attacks launched by collaborating malicious nodes whereas an individual node may not. However, in most cases only the local nodes can determine if incorrect information is included in a message.

All alarms are stored in a database with the witness' identity, source(s), destination(s), alarm type, and time stamp values. This information is then used by the reputation system to compute a reputation score for each node in the network.

E. *Computing Local Node Trust Values*

Our system includes negative feedback and forgiveness mechanisms in the trust metric calculation. The policy definitions include a *reputation decrease value* for all of the alarms generated by the detection system. Nodes are assumed to be trusted in our system and start with a reputation value of one. The computation for a node's reputation is one minus an exponentially weighted sum of the reputation decrease value for every alarm detected by the node locally (as shown in algorithm 1). This equation forgives nodes for past misbehavior over time and when an alarm is 24 hours old, it is removed by both the local node and the central authority. The exponential weight of 0.82 was selected to give a smooth weighting throughout the 24 hour period. This exponential weight will have a value of 0.008 after 24 hours thus approching a value close to zero.

Algorithm 1 Local computation of i's trust score

```
$K(i) \leftarrow 0$
**for all** $a$ such that $alarms \rightarrow source = i$ **do**
    $K(i) + = 0.82^{floor(hour(a->timestamp))} * a \rightarrow ReputationDecrease$
**end for**
$t(i) \leftarrow 1 - K(i)$
$s(i) \leftarrow max(t(i), 0)$
```

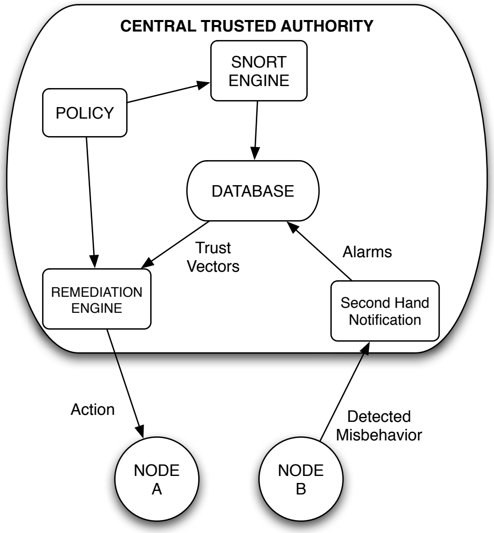

Fig. 3. Diagram of centralized trusted authority

The trust vectors from the nodes in the system are combined into a matrix M, where M_{ij} represents i's trust in node j. The centralized authority's trust vector is stored in a vector $\vec{c}$, where c_i is the centralized authority's trust in node i.

F. Computing Global Trust Values

As part of the computation of the global trust vector, the centralized authority's local trust vector $\vec{c}$ must be normalized in some manner. We define the normalized centralized authority's local trust vector $\vec{p}$ in the same way as described in EigenTrust.

$$p_i = \frac{\max(c_i, 0)}{\sum_j \max(c_j, 0)} \tag{1}$$

This value may be undefined if $\sum_j \max(c_j, 0) = 0$. We deal with this by setting $\vec{p}$ equal to a vector full of zeros. This normalization technique is not perfect in that the trust scores c_i and c_j could be equal and have a mediocre value. However, this normalization provides us a way to determine how much trust the centralized authority has in an individual node relative to the rest of the nodes in the network.

The centralized authority computes a global trust vector by combining its local trust vector $\vec{c}$ and all the trust vectors from the other members of the network stored in M. The centralized authorities trust vector $\vec{c}$ is multiplied by a weight a with a value $0 \leq a \leq 1$. The trust matrix of M is multiplied by the normalized form of the centralized authority's local trust vector $\vec{p}$, which is the centralized authority's relative trust in the node i. This is intuitive since the trust vectors from the nodes that the centralized authority trusts more should have a higher weighting in the global trust vector. If the trust vectors are from highly trusted nodes, one can reason that they should be less likely to be falsified. The final global trust vector is computed using equation 2.

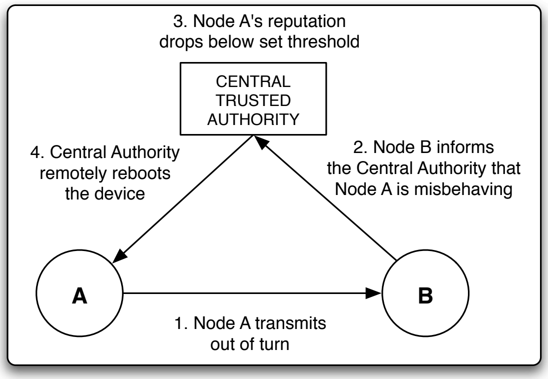
3. Node A's reputation drops below set threshold
CENTRAL TRUSTED AUTHORITY
4. Central Authority remotely reboots the device
2. Node B informs the Central Authority that Node A is misbehaving
A
B
1. Node A transmits out of turn

Fig. 4. Example remediation scenario

$$\vec{g} = a\vec{c} + (1-a)\vec{p^T} M \tag{2}$$

It would be dangerous to have the individual nodes in the system only use the global trust vector when determining if a message should be trusted. First hand dealings with a node should have substantial weight in future trust. Furthermore, a node could "game" the system by bad mouthing a single node while behaving properly to all the other nodes. In order to prevent this targeted malicious attack, the local nodes use a mixture of the global trust vector and their local trust vector as shown in equation 3. This equation multiplies g_i by the weight b and multiplies its local trust of i, t_i by 1 minus b, and then adding the two to achieve a trust metric r_i for the node i.

$$r_i = bg_i + (1-b)t_i \tag{3}$$

G. Remediation

The remediation system is an important part of the MIND system as it serves to enforce the policies of the network. The centralized authority is the only node with the authority to remediate other nodes in the system, thus remediation attempts by other nodes are ignored. The XML policy includes a number of tags that define different remediation actions to be taken by the centralized authority based on the current global reputation value g_i of the node i. Listed in order of severity these tags include Notify, Reboot, RadioSilence, and Remove. An example scenario showing how the remediation engine works is illustrated in Figure 4. In simple terms, we have programmed devices to respond appropriately to the given remediation command. The details of this remediation are beyond the scope of this paper.

The remediation system was designed to give misconfigured and misbehaving nodes a chance to reform and only remove malicious nodes from the system. The Reboot tag is meant to possibly reset misconfigured nodes and the Notify and RadioSilence remediation tags are designed to give misbehaving nodes a warning.

IV. Evaluation

To evaluate the MIND system, we setup an 802.11 wireless infrastructure network with ten authenticated members and one centralized trusted authority. To generate traffic we programmed nodes to send bursts of packets to a random destination. In order to simulate conversations, the receiver responds to a sender's packets with a reply message. The traffic generator for well-behaving nodes was programmed to ensure that the node did not violate the traffic limit policy of the network and that it adhered to other protocol policies 99% of the time. The traffic generator was configured to send on average 300 messages during a 30 minute interval. The traffic limit policy was set to 400 messages per 30 minute interval.

The policies for the network in our experiments were configured such that misconfigured nodes were given a chance to reform before being removed from the system. Thus it took longer to remove malicious nodes. The policies could be defined such that the network is more aggressive at removing nodes. This would remove malicious nodes from the system faster; however, this would also cause some misconfigured nodes to be removed before they are given a chance to reform their behavior. We set our threshold policies to remove malicious nodes within a 60 minute interval. This was an arbitrary selection and could have been set at 60 seconds.

A. Performance

Overall, the MIND system is extremely lightweight with the centralized authority responsible for the majority of the work. The message overhead of the MIND system was minimal and consisted of the centralized authority sending ten messages (one for every peer in the system) every five minutes to distribute the global trust vector. These messages are small in size consisting of only 20 bytes of payload data plus additional protocol overhead. The secondhand reports are also compact (4 bytes of payload). The trust computations are lightweight, with the bulk of the computational work being performed by the centralized authority, while the nodes are only required to combine the global and local reputation vectors.

B. Mixed Misconfigured and Malicious Nodes

In the first attack scenario, we configured two nodes to behave as if they were misconfigured and two nodes to behave maliciously. The misconfigured nodes sent messages with bad checksums until ordered to reboot by the centralized authority. The malicious nodes violated a number of network policies and did not comply with the centralized authority's remediation requests.

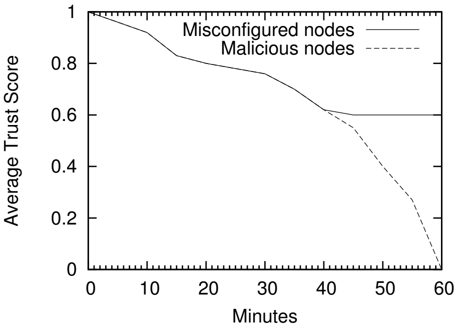

Fig. 5. Mixed misconfigured and malicious nodes

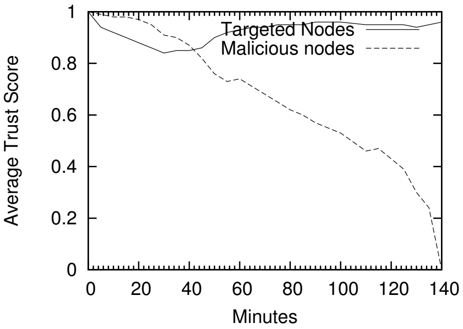

Fig. 6. Three malicious nodes launch a bad mouthing attack against three correctly behaving nodes

As Figure 5 shows, the malicious nodes where detected and when remediation attempts were ignored they were removed from the network after 58 minutes. The misconfigured nodes were instructed to reboot after 43 minutes and began to behave correctly. Once the misconfigured nodes corrected their behavior, their reputation began to increase through the forgiveness mechanism implicit in the trust computation. This shows that the MIND system can correctly distinguish between misconfigured nodes that correct their behavior and malicious nodes that continue to misbehave (and therefore must be removed from the network).

C. Bad Mouthing Attack

A common attack against reputation systems is for one or more colluding nodes to disseminate *false negative* information about honest nodes in order to decrease the honest node's reputation. This attack was studied in [21] and referred to as a "bad mouthing attack."

To analyze how MIND handles bad mouthing attacks, we configured the network to have seven well-behaving nodes and three colluding malicious nodes. These colluding nodes were configured to launch a bad mouthing attack against three honest nodes. The malicious nodes were also configured to violate network policies and provided bad information 50% of the time to the targeted honest nodes and to ignore all remediation instructions from the centralized authority.

As the results in Figure 6 show this test ran for two hours and 18 minutes before all malicious nodes where detected and

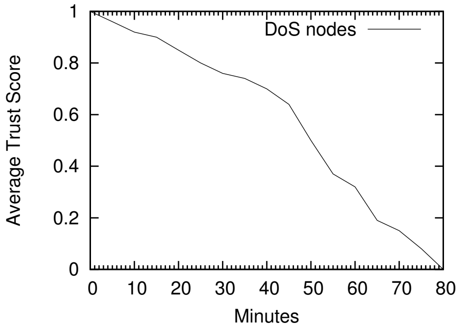

Fig. 7. Three nodes launch a DDoS attack

reputation scores reached the threshold to be removed from the system. At first, the bad mouthing attack was partially successful at lowering the reputation of the honest nodes. However, as the centralized authority detected the misbehavior of the malicious nodes, it began to discount the secondhand reports from these malicious nodes. Since the centralized authority did not detect any firsthand misbehavior from the correctly behaving nodes, it began to weight the correctly behaving nodes secondhand reports higher than the malicious node's reports. This resulted in the bad mouthing attack failing and the malicious nodes being removed from the system.

The weighting of secondhand reports by the firsthand information of the centralized authority proved to be effective even though the centralized authority could not detect all of the malicious node's misbehavior.

D. DDoS Attack

For our final experiment, we analyzed how the MIND system dealt with a number of colluding nodes launching a Distributed Denial of Service (DDoS) attack. We configured three nodes to behave maliciously by violating the message limit policy of the network. The nodes attempted to evade detection by other nodes in the network by only sending 399 messages during a 30 minute interval to every other node in the system. This resulted in the malicious nodes sending a total of 2793 messages per a half hour. Since the malicious nodes did not send more than 399 messages to any individual member of the network, no secondhand reports of DoS attacks were sent to the centralized authority. However, as shown in Figure 7, the centralized authority witnessed the attacks firsthand and the reputation of the malicious nodes fell sufficiently that they were removed from the network within one hour and seventeen minutes.

E. Evaluation Conclusions

These results show that the hybrid detection mechanism is effective at detecting attacks; arguably more effective than a purely centralized or purely distributed mechanism. Also, our evaluation shows that the trust computation can resist active attacks against the MIND system by correctly weighting secondhand reports based on firsthand knowledge of the node. As we previously indicated, the policies can be altered to allow the system to respond more quickly to misbehaving nodes. For example, we could have set the policy to remove a malicious node after five minutes instead of waiting for 58 minutes.

V. Related Work

Several representative P2P and wireless reputation systems currently exist. The EigenTrust and NICE systems focus on creating a reputation system for peer-to-peer networks. The Ebay trust scores are meant to facilitate trust between buyers and sellers on an e-commerce Internet site. The DISAS system is targeted at creating a secure routing protocol for multi-hop wireless networks.

Most online auction sites have implemented a reputation system. By far the most popular and widely used trust-management system is eBay's user feedback system. It uses a centralized server to collect feedback and display trust scores of other users. The data is readily available to the public and aids buyers and sellers in determining when to trust someone. An interesting analysis of eBay's reputation system was done by P. Resnick and R. Zeckhauser [22].

The NICE scheme is a trust inference mechanism targeted for peer-to-peer networks [8]. There are two components to the trust inference: local knowledge from past dealings and a mechanism to query other trusted members of the system for secondhand information about a node. The system also includes virtual currency in the form of cookies generated after successful transactions to record direct trust between peers.

Sonja Buchegger and Jean-Yves Le Boudec designed a distributed reputation system [6], in which the secondhand reputation rating is accepted only when it is close to the current reputation rating. The system has a stipulation that secondhand information that deviates too much from the current rating is ignored; thereby improving the robustness of the system against bad mouth attacks and reputation inflation by malicious nodes.

The EigenTrust algorithm [16], uses both positive and negative firsthand and secondhand feedback to capture a peer's reputation. The algorithm computes a score by taking the number of positive transactions divided by the total number of transactions with the peer. The scores are normalized over all peers in the system to provide the relative trust of every node in the network. EigenTrust is fully distributed using a DHT-overlay network. The MIND algorithm for computing trust scores borrows from equations described in the EigenTrust paper.

The DICAS protocol [12], is a lightweight method for detecting malicious nodes in multi-hop wireless networks. It can detect wormhole, Sybil, and rushing attacks from malicious nodes in the network. It also includes a secure neighbor discovery and authentication algorithm. DICAS also defines LSR (Lightweight Secure Routing), which is built on top of the DICAS protocol and is meant to enable nodes to route around malicious nodes. The MIND system assumes a small single-hop network of authenticated nodes and therefore avoids the multi-hop routing issues.

None of the systems mentioned above discuss the policy aspects of defining rules and consequences for a wireless network. Much of what we developed in MIND focuses on methods for defining policies and consequences for violating these policies. Moreover, all of the protocols above assume a fully distributed network with no centralized authorities. We believe that certain networks deployed for mission critical applications will include a centralized structure with trusted authorities. The MIND system approaches the problem from this perspective.

VI. Conclusion

In this paper, we described a method for detecting and responding to misbehaving nodes in a single-hop wireless network. This method makes use of a reputation-based intrusion detection system in which a centralized trust authority monitors traffic and also collects secondhand information on potentially misbehaving nodes. We show that the hybrid detection approach, which blends centralized and distributed behavior monitoring, is able to detect a variety of attacks. We also demonstrate that the trust computation can resist active attacks by correctly weighting secondhand and firsthand knowledge about the nodes in the network. In evaluating the system, we demonstrate that it can detect and respond to misconfigured, malicious and bad mouthing nodes as well as to DDoS attacks. As for future work, we are presently testing a kernel level detection mechanism that will provide physical layer reports from the various nodes to the central authority. We hope to use this capability to instantiate a more trustworthy mechanism for reporting secondhand information.

References

[1] J. Steinheider, V. Lum, and J. Santos, “Field trials of an all-software GSM base station,” in *Software Defined Radio Technical Conference*, November 2003.

[2] J. Brucker-Cohen, “WIFIHOG,” Internet http://www.mee.tcd.ie/~bruckerj/projects/wifihog.html, 2006.

[3] R. L. Rivest, A. Shamir, and L. M. Adelman, “A METHOD FOR OBTAINING DIGITAL SIGNATURES AND PUBLIC-KEY CRYPTOSYSTEMS, Tech. Rep. MIT/LCS/TM-82, 1977. [Online]. Available: citeseer.ist.psu.edu/rivest78method.html

[4] A. Lockhart, “Wireless snort ids,” Internet http://snort-wireless.org/, 2006.

[5] “Snort intrusion detection and prevention system,” Internet http://www.snort.org/, 2006.

[6] S. Buchegger and J.-Y. L. Boudec, “A robust reputation system for p2p and mobile ad-hoc networks,” in *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems.*, 2004. [Online]. Available: citeseer.ist.psu.edu/buchegger04robust.html

[7] M. Gupta, P. Judge, and M. Ammar, “A reputation system for peer-to-peer networks,” in *NOSSDAV, June*, 2003. [Online]. Available: citeseer.ist.psu.edu/gupta03reputation.html

[8] S. Lee, R. Sherwood, and B. Bhattacharjee, “Cooperative peer groups in nice,” in *INFOCOM, Apr.*, 2003. [Online]. Available: citeseer.ist.psu.edu/lee03cooperative.html

[9] L. Xiong and L. Liu, “Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities.” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, 2004.

[10] H. Frey, J. Lehnert, and P. Sturm, “Ubibay: An auction system for mobile multihop ad-hoc networks,” in *Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments (AdHocCCUCE’02)*, 2002. [Online]. Available: citeseer.ist.psu.edu/frey02ubibay.html

[11] S. Ganeriwal and M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” in *SASN ’04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM Press, 2004, pp. 66–77.

[12] I. Khalil, S. Bagchi, and C. Nina-Rotaru, “Dicas: Detection, diagnosis and isolation of control attacks in sensor networks,” in *In the IEEE Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm) Athens, Greece September 5 - 9*, 2005.

[13] P. Michiardi and R. Molva, “Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks,” in *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*. Deventer, The Netherlands, The Netherlands: Kluwer, B.V., 2002, pp. 107–121.

[14] G. Shafer, *A mathematical theory of evidence*. Princeton University, 1976.

[15] A. Jøsang and R. Ismail, “The beta reputation system.” 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy, June 2002.

[16] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in p2p networks,” in *WWW ’03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2003, pp. 640–651.

[17] T. Bray, J. Paoli, and C. M. Y. cQueen, “Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation,” Internet http://www.w3.org/TR/2004/REC-xml-20040204, February 2004.

[18] M. Marchiori, “The Platform for Privacy Preferences 1.0 (P3P1.0) specification,” Internet http://www.w3.org/TR/2002/REC-P3P-20020416/, April 2002.

[19] T. Moses, “eXtensible Access Control Markup Language (XACML) version 2.0, OASIS standard,” Internet http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, February 2005.

[20] N. GENERATION, “XG,” Internet http://www.darpa.mil/ato/programs/XG/, 2005.

[21] C. Dellarocas, “Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems,” in *ICIS*, 2000, pp. 520–525. [Online]. Available: citeseer.ist.psu.edu/455920.html

[22] P. Resnick and R. Zeckhauser, “Trust among strangers in internet transactions: Empirical analysis of ebay’s reputation system,” in *The Economics of the Internet and E-commerce, M.R.Baye, ed., Elsevier, 2002, pp. 127.157.*