Reliability Estimation in Mobile Ad hoc Networks

Ritu Chadha, Alexander Poylisher, Constantin Serban
Applied Communication Sciences
150 Mount Airy Road, Basking Ridge, NJ 07920, USA
{rchadha, apoylisher, cserban}@appcomsci.com

Abstract—**We describe a system for estimating the reliability of nodes in a mobile ad hoc network. The goal of the system is to detect insider attacks on the control plane of wireless protocols at the link and network layers, and to generate and propagate corresponding reliability estimates for nodes across the network. Our contributions are two-fold: first, we implement a cross-layer invariant-based technique for attack detection, where rules about correct combined behavior of protocols are specified based on data collected from multiple layers of the protocol stack. Next, we use the results of our attack detection techniques to compute reliability estimates for network nodes, where a reliability estimate represents the estimate of a node *N* for another node *M*. The form of our reliability estimate for a node is expressed as (*t*,*c*), where *t* indicates the trust for that node, and *c* indicates the confidence in this trust value. Reliability estimates are propagated across the network in a manner that is resilient to malicious nodes that propagate false reliability estimates, and minimizes network overheads. Our simulation results show that the above techniques result in highly accurate reliability estimates even in the presence of multiple malicious nodes in the network.**

Keywords: Attack detection, control plane attacks, trust propagation, intrusion detection, mobile ad hoc networks.

I. Introduction

Much of the work that has been performed in the area of Mobile Ad hoc Network (MANET) technologies in the recent past has focused on improving performance and maximizing network throughput under benign conditions. However, as various studies have shown, the control plane protocols used in these networks for coordinating activities at various layers of the stack are often vulnerable to attacks. The problem is even more pronounced when the attacker is an "insider" and has access to a network node, where he or she can modify control plane software at will and thereby disrupt network operations. Wireless control protocols share information about network resources such as spectrum, power, link state, routing, etc. Based on the exchanged information, nodes make decisions about frequency usage, data transmission schedules, how to route network traffic, and so on. A malicious node can wreak havoc on the network by manipulating the information transmitted via these protocols.

In order to enable network nodes to protect themselves against network attacks, they must be able to detect these attacks and develop reliability estimates for other network nodes. Mitigation techniques can then be developed to route around or otherwise minimize the impact of malicious nodes. The scope of the work described in this paper is not to develop these mitigation techniques, but rather, to develop techniques for accurate estimation of the reliability of network nodes, to provide a basis for deciding about potential mitigation techniques. The challenges to be addressed are two-fold: first, how do we accurately detect attacks against wireless control protocols, and second, how do we propagate information about detected attacks to other nodes so that nodes can accurately estimate the reliability of nodes in their vicinity? These are the challenges that we address in this paper.

In this paper, we describe a *cross-layer invariant-based* technique for attack detection, where rules about correct *combined* behavior of protocols are specified based on data collected from multiple layers of the protocol stack. The use of invariants within a single protocol has been shown [5] to detect a variety of possible malicious behavior with very few basic rules concerning the single protocol. In contrast, our approach works *across multiple protocols and layers of the protocol stack* and specifies events that must occur in all of these layers if the protocol stack operates correctly. Next, we describe a method for computation of unified reliability estimates that addresses the challenge of computing such estimates in a highly dynamic and bandwidth-constrained wireless network where many nodes may be unreliable or malicious. Our approach is partly based on [17], where unified reliability estimates are computed using a very efficient mathematical approach that is tailored for such environments and has been shown to be very robust against lying nodes.

The outline of the paper is as follows. In Section II, we describe related work. In Section III, we provide an overview of our approach, and in Section IV, we present our system evaluation. We summarize our conclusions and discuss future work in Section V.

II. Related Work

There are three major existing categories of network attack detection systems [2]: anomaly detection systems that are trained to recognize normal behavior using machine learning techniques; misuse detection techniques that store patterns, or signatures, of known attacks and compare them with observed data; and specification-based detection, where the correct operation of a protocol is specified and its execution is monitored to verify that it conforms to the specification. It is well known that anomaly detection techniques provide the ability to detect zero-day attacks and generally provide excellent attack detection capabilities (or low rates of false negatives), but typically have a high rate of false positives (false alarms), whereas misuse-based and specification-based techniques are effective at detecting attacks with a low rate of false positives but a high rate of false negatives (undetected attacks) [3]. Our approach differs from past work because (i) it focuses on the detection of attacks against multiple control protocols, which is a relatively immature area, and (ii) our proposed cross-layer invariant-based detection has not been tried before; the closest work we are aware of specifies invariants for single protocols [5].

Trust evaluation/analysis [8-13] and trust aggregation [4,6,8,11,13,14] have been studied in a general setting, as well as for MANETs specifically [14,16,17,18]. Relevant work can be partitioned into centralized approaches, that require every node to share trust information with every other node, and distributed aggregation approaches such as that described in [17]. Our approach is based on [17] and is superior to centralized approaches due to its bandwidth efficiency, which is critical for wireless networks. This is enabled by our trust computation operators that allow efficient local computation of trust based solely on the opinions of neighboring nodes, as well as the fact that we strictly limit the propagation of reliability estimates to a few hops and we do not propagate reliability estimates that have low confidence values. Our approach is also resilient to nodes lying about reliabilities.

III. Overview of Technical Approach

A. *Cross-Layer Invariant-Based Detector*

We have developed a cross-layer invariant-based detector, where rules about correct *combined* behavior of protocols are specified based on statistics collected at multiple layers of the protocol stack. This enables automation of the application of substantial available expert knowledge of current/future control protocols, tactical radios, platform capabilities and mission information, and makes our system easily extensible, since new invariants can be added as humans gain more insights and expertise in the correct operation of different control protocols. A critical point to note is that these invariants are independent of network conditions, and are therefore *robust to the dynamics of a wireless network*.

Our detector extracts data from packet and frame traces overheard from multiple layers of the protocol stack, including transport, network, and MAC, as well as associated link layer data (e.g., SNR). Each node maintains a near-term history of such traces, and invariants are evaluated on the conversations extracted from the traces. Finally, we leverage the fact that for certain network deployments, there is typically a lot of additional information available, such as asset information and capabilities, node movement capabilities, node locations, node mobility plans, etc. Our invariants use this additional information when relevant. Below, we present a few illustrative invariant examples for OLSR HELLO messages and IP forwarding. In addition to invariants about control protocols, we also specify invariants that catch attacks against the reliability propagation algorithm.

Info used	**Description of some sample invariants and what a violation may indicate**
OLSR and physical constraints, location data	For each neighbor node N claimed by node X in a HELLO message, an observer node Y verifies *the possibility of node N being X's neighbor against theoretical maximum ideal radio coverage and likely current coordinates, given most recently received actual coordinates/direction of motion and possible node movement speeds.* If the number of inconsistencies is above a specified ratio threshold, Y places X under suspicion.
OLSR, MAC	Nodes X and Y are one-hop neighbors and X is the next hop from Y to Z. Y has X's most recent advertised neighbor set and continuously observes MAC layer transmissions from X. *When Y sends traffic destined to Z, verify that X forwards Y's messages.* If not, Y places X under suspicion.

After all necessary invariants have been evaluated, the results are used to create numerical trust values for the neighbors of the local node; e.g., a node that has been identified as a definite violator of invariants is assigned a trust value of 0, and a node that may be a violator is assigned a trust value proportionate to the certainty associated with the diagnosis. The associated confidence value is computed based various factors including the quality and quantity of the input data provided to the detector.

B. *Reliability Computation*

The form of our reliability estimate for a node is expressed as (t,c), where t indicates the *trust* for that node, and c indicates the *confidence* in this trust value. A node derives *trust* estimates for its neighbors based on input from the attack detector described above. The *confidence* level is a measure of how “good” the trust estimate is, based on a variety of indicators, including the noisiness of the RF environment (which influences the correctness of the observable features), the availability of external information (such as location information), etc. Note that these indicators use information that is external to the detectors, and therefore cannot be captured within the trust value t of the reliability estimate.

The cross-layer detector described in the previous section is responsible for generating a node’s estimates of its neighbors’ reliabilities. However, each node in the network needs to know about the reliability of more than its one-hop neighborhood, including non-neighbor nodes that it cannot directly observe. In this section, we explain how a node computes reliability estimates for nodes that are not its neighbors by aggregating reliability estimates obtained indirectly via other nodes, using the algorithm shown in Figure 1. Our objective is to compute these estimates in a manner consistent with the following underlying intuitive principles: (1) trust information obtained second-hand is *discounted* based on the reliability of the supplier of the estimate; in particular, reliability estimates from nodes with very low trust are *discarded;* (2) when multiple trust opinions are obtained, estimates with lower confidence values are disregarded in favor of those with higher confidence values; and (3) when multiple opinions are obtained, they are compared for inconsistencies to detect lying nodes.

Reliability estimates can be combined in many ways that satisfy properties (1) and (2). We compute unknown trust and confidence values along a “trust path” (i.e., a virtual “path” from one node to another via zero or more nodes such that the trust of each node for the next along the path is known) as follows: two estimates (t_1,c_1) and (t_2,c_2) are combined using an operator $\otimes$ defined by: $(t_1,c_1)\otimes(t_2,c_2)=(t_1*t_2,c_1*c_2)$, and multiple opinions are combined by picking the one with higher confidence using operator ⊕ defined by: (t_1,c_1) ⊕ (t_2,c_2) is (t_1, c_1) if $c_1 > c_2$ and (t_2, c_2) if $c_2 > c_1$. These operators along with the set S of reliability estimates form an algebraic structure called a semiring [15]. The computation of reliability estimates (t,c) can be *performed extremely efficiently, in a distributed manner*, thanks to the distributive and associative properties of semiring operators. This allows combining local estimates of neighbors with neighbors' estimates of other nodes, rather than having to talk to every node to get its reliability estimate of every other node.

```
// Notation: $T_q[u]$ and $C_q[u]$ denote trust/confidence that q has for u
q ← this node;
if this is the first run of this algorithm then
    for each $u \in V$ do // initialize estimates at start
        $T_q[u] = 0.5$ // medium trust for all other nodes
        $C_q[u] = 0.1$ // low initial confidence
    $T_q[q] = C_q[q] = 1$ // perfect reliability for self
repeat at every decision period
    // First, use detectors' decisions for neighbors
    for each $v \in$ Neighbors[q] do
        invariant_detector($v, T_q[v], C_q[v]$);
    // Next, compute reliability estimates for all nodes
    for each $u \in V$ do
        if $u \neq q$ and $u \notin$ Neighbors[q] then
            // skip myself and neighbors
            // first, decay my current estimate for u
            // obtained from the previous round, if any
            trust = $T_q[u]$ * decay // 0 < decay < 1
            confidence = $C_q[u]$ * decay // 0 < decay < 1
            // the following loop combines estimates for u from
            // all my neighbors with my own estimates of them
            for each $v \in$ Neighbors[q] do
                if $T_q[v]$>min_trust // Only consult nodes that q trusts
                then (trust, confidence) = (trust, confidence)⊕
                        $((T_v[u],C_v[u]) \otimes (T_q[v],C_q[v]))$
        $T_q[u]$ = trust; $C_q[u]$ = confidence
```

Figure 1. Generalized shortest distance algorithm

We use a very efficient mechanism for disseminating REs to neighbors that relies on every node periodically broadcasting its local view of network reliability estimates to its neighbors (an aperiodic broadcast could be performed if a new malicious node is detected). Note that this hop-by-hop propagation mechanism requires multiple decision periods for information to propagate across network hops; however, we argue that this is not an issue because the further away two nodes are from each other, the less they care about each others' view of REs.

IV. System Evaluation

In this section, we describe our evaluation results, performed using a 50-node MANET simulation in ns-2.

A. *Experiment 1: Vary Traffic Volume*

We performed experiments to study accuracy of attack detection as traffic volume is varied, which impacts *attack observability*. We injected a black hole attack at random at different nodes in the network, with only one node attacking at any one time. The attack was implemented by having the malicious node advertise a number of fictitious neighbors through OLSR HELLO messages, and drop all packets that it should have forwarded.

We used invariants to detect this attack based on the premise that neighbors of the malicious node can observe, through promiscuous reception, messages received and sent by their neighbors. If a node is found not to forward the messages it receives for forwarding, it is identified as an attacker. In practice, however, the observing node may be unable to observe all the transmissions of neighbors, due to interference from other nodes, which needs to be taken into account. This is usually the case when the network is congested. We specified the following invariant to represent this condition (violation of this invariant indicates an attack):

$$h * \# \text{ pkts forwarded} / \# \text{ pkts received for forwarding} > \text{threshold} \quad (1)$$

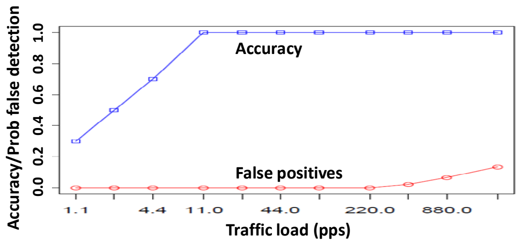

Figure 2. Traffic Volume Experiment

where *h* is a factor that represents a measure of channel contention, computed as the ratio of the total number of packets seen at the witnessing node and the sum of packets-to-forward and packets forwarded. This adjusting factor allows us to favor decisions made in a clean environment and discount decisions made when the vantage point is noisy. For our experiments, we used a fixed threshold of 0.5 and a decision period of 30 seconds. Each data point in the experiment was averaged over 10 runs. We used ns-2 to simulate a 50-node 802.11 ad hoc network. Our experiments measured detection accuracy (in terms of true positive and false positive rates) for different traffic volumes and different mobility regimes. We report an accuracy of 1 if the malicious node was correctly detected by at least one neighbor, for at least two decision periods out of three (the scenario was 3 decision periods.) The false positive rate was computed as the sum of the number of wrong decisions divided by the number of nodes. Figure 2 shows our results. The accuracy is lower when there is little traffic being sent, as the amount of traffic that can be dropped at a single node is low, thus reducing *attack observability*. Note that the black hole attack was detected with probability of 100% for several values of traffic load (see Figure 2). The probability of false alarm was <10% for all scenarios, except for one data point. Also, there was no network overhead as all statistics used were local.

B. *Experiment 2: Vary Number of Malicious Nodes*

We performed experiments to study how reliability estimate accuracy varies as the number of malicious nodes in the network is varied. This time, we injected the same black hole attack as before, at random, but gradually increased the number of malicious nodes in the network. We used the same invariant-based detector as in the previous experiment; this time, we computed the local trust value for a neighbor based on (1) above as follows:

$$\text{trust} = h * \# \text{ packets forwarded} / \# \text{ packets received for forwarding} \quad (2)$$

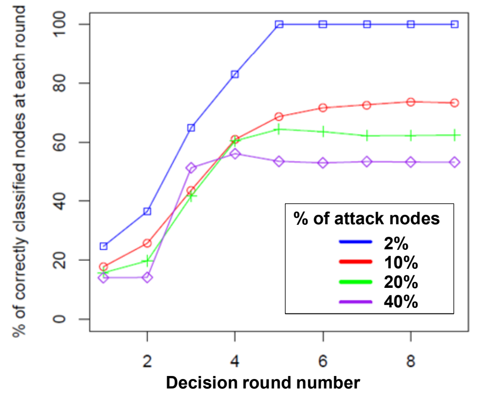

Figure 3. Accuracy vs. # malicious nodes

Each node computes reliability estimates (*t,c*) (*t=trust, c=confidence)* based on the algorithm in Figure 1 and broadcasts its estimates for other nodes to its MAC neighbors every 30 seconds. Bad nodes always broadcast reliability estimates of (0,1) for all other nodes. We repeated the broadcast twice per decision period, for redundancy; the resulting network overhead was computed to be 0.1% of network capacity, which was estimated to be 28.8Mbps. We again used our 50-node network. Each point on the graph in Figure 3 represents the percentage of correctly identified bad nodes at each round, or decision period. This is measured as the ratio "hitCount/(NN*CN)", where hitCount is the sum of the correctly identified compromised nodes at all the non-compromised nodes across the entire network, NN is the number of non-compromised nodes, and CN is the number of compromised (attacker) nodes. A non-compromised node is said to have correctly identified an attacker node if the trust measure of the target at the observer is <0.5.

Figure 3 shows 4 curves, corresponding to 2%, 10%, 20%, and 40% of the nodes being compromised and launching a black hole attack. Note that each curve flattens out when the number of rounds roughly corresponds to the diameter of the network (5 hops), by which time each node in the network is able to derive a non-default trust value for all nodes in the network.

Note that the black hole attack was detected correctly by at least 7 nodes at the first decision round (in the worst case where 40% of the nodes were bad), and an increasing number at each subsequent round until convergence at 4-5 rounds. The network overhead was 0.1% of network capacity for this case.

Experiments 1 and 2 together show that black hole attacks can be detected with very high accuracy and a low number of false positives when there is a small number of malicious nodes in the network; this accuracy degrades gradually as the number of malicious nodes increases, as shown in Figure 3. Also, our reliability estimation approach is shown to be resilient against lying nodes (Figure 3) and almost all "good" nodes eventually arrive at a correct assessment of the "bad" nodes.

V. Conclusions and Future Work

We have developed a robust reliability estimation system based on a cross-layer invariant-based detector that looks for behavior that violates rules about correct cross-layer behavior of control plane protocols, and a resilient reliability computation and propagation scheme that is robust to "bad mouthing" attacks, where nodes lie about their reliability estimates. Our reliability propagation scheme poses very low overheads, which is critical in a MANET where bandwidth is scarce. The next step for this research is to develop mitigation techniques that make use of our reliability estimates to determine how to maintain good network performance in spite of the presence of malicious nodes in the network.

VI. References

[1] Airmon-ng. "Monitor mode on wireless interfaces" (online: http:// www.aircrack-ng.org/doku.php? id=airmon-ng), last acc. May 2013.

[2] Tiranuch Anantvalee, Jie Wu. "A Survey on Intrusion Detection in Mobile Ad Hoc Networks". Wireless/Mobile Network Security, Y. Xiao, X. Shen, and D.-Z. Du (Eds.), pp. 170 – 196, Springer, 2006.

[3] F. Anjum, D. Subhadrabandhu, S. Sarkar. "Signature based Intrusion Detection for Wireless Ad-Hoc Networks: A Comparative study of various routing protocols". IEEE 58th Veh. Tech. Conf. (VTC), 2003.

[4] Y Bachrach, A. Parnes, A. Procaccia and J. Rosenschein, "Gossip-based Aggregation of Trust in Decentralized Reputation Systems". Auton Agent Multi-Agent System (2009) 19:153-172.

[5] Frédéric Cuppens, Nora Cuppens-Boulahia, Seila Nuon, Tony Ramard. "Property Based Intrusion Detection to Secure OLSR". 3rd Intl. Conf. on Wireless and Mobile Communications (ICWMC '07), 4-9 March 2007.

[6] S. Foley, W. M. Adams, B. O'Sullivan. "Aggregating Trust Using Triangular Norms in the KeyNote Trust Management System". Security and Trust Management, LNCS, Vol. 6710, 100-115, 2011.

[7] K. Govindan, P. Mohapatra. "Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey". IEEE Communications Surveys & Tutorials, Volume 14, Issue 2, pp. 279–298, 2012.

[8] Rolf Haenni and Jacek Jonczy. "A New Approach to PGP's Web of Trust". ENISA European eIdentity conference, Paris, 2007.

[9] J. Huang and D. Nicol. "A Calculus of Trust and its Application to PKI and Identity Management". In Proc. of IDTrust, 2009.

[10] Bert Huang, Angelika Kimmig, Lise Getoor, Jennifer Golbeck. "Probabilistic Soft Logic for Trust Analysis in Social Networks". International Workshop on Statistical Relational Artificial Intelligence (StaRAI), 2012.

[11] A. Jøsang. "An algebra for assessing trust in certification chains". In Proceedings of the Network and Distributed Systems Security Symposium (NDSS), 1999.

[12] Audun Jøsang, Tanja Azderska, Stephen Marsh. "Trust Transitivity and Conditional Belief Reasoning". 6th IFIP WG 11.11 International Conference on Trust Management, Surat, India, pp. 68-83, 2012.

[13] S. D. Kamvar, M. T. Schlosser and H. Garcia-Molina. "The Eigen-Trust algorithm for reputation management in P2P networks". In Proc. of the 12th Intl. Conference on World Wide Web, pp. 640–651, May 2003.

[14] Yan Lindsay Sun, Wei Yu, Zhu Han, and K. J. Ray Liu. "Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks". IEEE JSAC, Vol. 24, No. 2, Feb. 2006.

[15] M. Mohri. "Semiring frameworks and algorithms for shortest-distance problems". Journal of Automata, Languages, and Combinatorics, vol. 7, no. 3, pp. 321–350, January 2002.

[16] Yonglin Ren and Azzedine Boukerche. "Modeling and Managing the Trust for Wireless and Mobile Ad hoc Networks". In Proceedings of the IEEE International Conference on Communications (ICC), 2008.

[17] G. Theodorakopoulos, J. S. Baras. "On Trust Models and Trust Evaluation Metrics for Ad-Hoc Networks". *Journal of Selected Areas in Communications, Security in Wireless Ad-Hoc Networks,* Vol. 24, Number 2, pp. 318-328, February 2006.

[18] G. E. Theodorakopoulos. "Robust Network Trust Establishment for Collaborative Applications and Protocols". Ph.D. Dissertation, U. Maryland, 2007.