RIDA: Robust Intrusion Detection in Ad Hoc Networks

Dhanant Subhadrabandhu[⋆], Saswati Sarkar, Farooq Anjum

Abstract. We focus on detecting intrusions in wireless ad hoc networks using the misuse detection technique. We allow for detection modules that periodically fail to detect attacks and also generate false positives. Combining theories of hypothesis testing and approximation algorithms, we develop a framework to counter different threats while minimizing the resource consumption. We obtain computationally simple optimal rules for aggregating and thereby minimizing the errors in the decisions of the nodes executing the intrusion detection software (IDS) modules. But, we show that the selection of the optimal set of nodes for executing the IDS is an NP-hard problem. We present a polynomial complexity selection algorithm that attains a guaranteeable approximation bound. We also modify this algorithm to allow for seamless operation in time varying topologies, and evaluate the efficacy of the approximation algorithm and its modifications using simulation. We identify a selection algorithm that attains a good balance between performance and complexity for attaining robust intrusion detection in ad hoc networks.

1 Introduction

Ad hoc networks provide the only means of electronic communication in areas where establishing infrastructure like base stations is either impossible or not cost-effective. Examples include disaster recovery operations, battlefields, communication in remote terrains (e.g., reservations, rural areas), events like superbowl matches, etc. Ad hoc networks do not need infrastructure because users perform some communication tasks like relaying packets. But, if the network is to provide any quality of service (QoS) guarantee it can utilize the users but can not solely rely on them. This is because users may be available for short durations only. The QoS guarantees can however be provided if some easily deployable low complexity system nodes e.g., static and mobile access points are available. These nodes together with users who are trusted by the network and are in the network most of the time can be relied upon for relaying packets, discovering routes, securing the communication, etc. Thus, there are two sets of nodes: users who only communicate using the network but do not perform system tasks (outsider nodes), and static and mobile access points[⋆⋆] and users that communicate and perform network tasks (insider nodes). As an example, consider a wireless network in a university. Universities have static access points deployed in several places. Mobile access points carried by personnel and vehicles can provide additional coverage as required, e.g., in areas of heavy network traffic like games, concerts, etc. This motivates the utilization of ad hoc networks in universities. In such ad hoc networks, the terminals of the university employees can also perform system tasks as required. The access points and the employees' terminals constitute insider nodes. Students as well as other visitors constitute outsider nodes. As another example, a disaster recovery team can use ad hoc networks to provide services like email, news, audio/video applications etc. in an area where

[⋆] D. Subhadrabandhu and S. Sarkar are at the Electrical and Systems Engineering Department of the University of Pennsylvania. F. Anjum is at Telcordia Technologies. The contribution of S. Sarkar was in part supported by the National Science Foundation under grants ANI-0106984, NCR-0238340 and CNS-0435306.

[⋆⋆] These access points operate in the ad hoc mode and not in the infrastructure mode of 802.11 networks.

communication infrastructure has been damaged due to a natural disaster or terrorist activity. The insider nodes are small access points on buildings and mobile terminals carried by the personnel. The outsider nodes are civilians who communicate with each other using the network. We postulate that ad hoc networks deployed in near future will match this description. We address security threats in these.

These networks will be used by a diverse user population, e.g., civilians in disaster hit areas, students in universities etc., which increases the security risks. One such risk is a user who subverts the functioning of the network by causing undesirable events. Such users are considered as intruders and the events as intrusions. Examples of intrusions are attacks such as TCP SYN flood*, Land Exploit**, SSPing*** etc. [1] [2]. These intrusions leverage system vulnerabilities. There are two ways to prevent such intrusions. One way is to remove the vulnerabilities from the system such as by designing resistant protocols like SCTP [3] to resist TCP SYN flood attacks, patching the operating systems, etc. But, this may not be possible due to various reasons such as poor design [4], limited use of efficient technical solutions (e.g., SCTP is rarely used due to large scale deployment of TCP), different devices having different capabilities, inefficient configuration (e.g., users do not change default security settings or apply patches), etc. The second approach, which is complimentary to the first, is to detect attempts to leverage the vulnerabilities and stop such attempts from succeeding. In this paper, we focus on the second approach which is referred to as intrusion detection.

Intrusion detection has been extensively investigated for wireline networks [5], [6]. But techniques geared towards wireline networks would not suffice in an ad hoc network due to the ease of listening to wireless transmissions, lack of fixed infrastructure, etc. [7]. For example, several detection strategies in wireline networks are based on the presence of a small number of gateways that route and therefore monitor all traffic. But, ad hoc networks typically do not have such choke points and even if such choke points exist, their locations continuously change due to mobility. Thus, designing the optimal selection strategy is more complex in ad hoc networks. Also, intrusion may be detected in wireline networks by detecting anomaly, i.e., by comparing the current system behavior with that in absence of intrusion. In ad hoc networks, however, normal behavior can not be accurately characterized, e.g., a node may transmit false updates since the routing protocol is slow to converge and not because it is malicious. Further, unlike in wireline networks, nodes in an ad hoc network have limited energy. Hence, only computationally simple, energy-efficient detection strategies can be used. The detection algorithms must also be distributed as communication with a central computing unit will consume significant energy. Finally, the detection algorithms must seamlessly adapt to topological changes due to mobility.

A detection strategy specifically suitable for ad hoc networks is that of misuse detection that relies on the use of known patterns of unauthorized behavior [8]. More specifically, this technique detects intrusion when the transmitted traffic contains abnormal packets which serve as "signatures" of attacks. For example, the signature of SSPing attack is a series of highly fragmented, oversized ICMP packets. A SYN packet with the same source and destination address indicates a land exploit attack [1]. The advantage of this technique in ad hoc networks is that it does not require characterization of normal behavior. This technique can not however detect attacks whose signatures are unknown.

The misuse detection technique requires some nodes to capture (sniff) and analyze the network traffic. Therefore, a prerequisite for deploying misuse detection in ad hoc networks is to determine which nodes should execute the sniffing and analysis software modules which we refer to as the

* The attacker opens a large number of half-open TCP connections.

** The attacker sends a TCP SYN packet with the same target and source address.

*** The attacker sends a series of highly fragmented, oversized ICMP packets.

intrusion detection software (IDS) modules. Previous works have considered this problem assuming that the insider nodes that analyze the traffic detect malicious packets without any failure [8], [9], [10]. But some insiders periodically stop functioning because of operational failure and low residual energy and would not detect attacks during those intervals. Insider nodes may also erroneously conclude intrusion when there is none. It is difficult to quickly detect which insider nodes have failed, and attacks may be deliberately launched before the IDS can be activated in other nodes.

We focus on the misuse based detection problem in ad hoc networks while allowing for erroneous decisions made by insider nodes executing the IDS modules. We describe our system model in Section 2. Combining theories of hypothesis testing and approximation algorithms, we develop a framework to counter different threats while consuming the minimum possible resource (Section 3). We obtain computationally simple optimal rules for aggregating and thereby minimizing the errors in the decisions of the nodes detecting the intrusion. But, we prove that optimally selecting the nodes for sniffing and analyzing packets is NP-hard. Then, we present an approximation algorithm APPROX that attains a guaranteeable approximation bound in polynomial complexity. We also modify APPROX to ensure seamless operation in time varying topologies. Using simulations, we evaluate the detection costs and security risks of different algorithms and identify one selection algorithm MUN that attains a good balance between performance and complexity for attaining robust intrusion detection in ad hoc networks (Section 4). Refer to technical report [11] for proofs. Details related to implementation of the proposed schemes are beyond the scope of this paper.

2 System Model

A network consists of two types of nodes: insider nodes and outsider nodes. An outsider node may launch attacks on another outsider or an insider node. Therefore, any outsider node that sends traffic is potentially malicious, and is referred to as an intruder. There may be multiple intruders who may use any set of paths for transferring their packets. The number and the locations of the outsider nodes and their destinations are not known to the network, and vary with time. We assume that each attack consists of one packet, e.g., a land exploit attack [1] consists of one packet. Packets constituting attacks are denoted as *bad* packets. A packet not constituting an attack is denoted as a *good* packet.

We represent a wireless network by an undirected graph $G(V, E)$. Here, $V = \{1, \ldots, N\}$ consists of the insider nodes and E is the set of edges between the insider nodes. There exists an undirected edge between any two insider nodes which are in each others' transmission range.

Definition 1. *A neighborhood N_i of an insider node i is the set of insider nodes that are within i's transmission range. An insider node i covers every insider node in its neighborhood.*

Thus, an insider node is always its own neighbor and covers itself.

Insider nodes execute the IDS modules that employ misuse-based detection strategy so as to detect bad packets while in transit between the intruder and the destination. Some insider nodes may not have the capability to execute the IDS. Thus, insider nodes are of two types: (a) *IDS capable* and (b) *IDS incapable*. Also, different IDS capable insider nodes e.g., PDAs, laptops, access points etc. consume different amount of resources to execute the IDS, since they have different residual energy and computational capability. An IDS capable insider node i has weight w_i that represents its resource consumption when it executes the IDS. Depending on the system policy, some but not all the IDS capable insider nodes will execute the IDS - these are denoted as *IDS active*. The set of IDS active nodes may dynamically change depending on available bandwidth, computational resources, energy and the current topology.

An IDS active insider node operates in promiscuous mode, i.e., receives any packet that is transmitted by any of its neighbors. Several authors e.g. [8],[12], assume operation in promiscuous mode given its advantages (e.g., neighborhood monitoring).

An IDS active insider node may sporadically fail to detect bad packets and report good packets as bad. A node may not detect bad packets during power saving operations, or if the attack has been designed to evade its IDS module [13],or if the IDS modules on the node are out of date, or if the attacker successfully launches a denial of service (DOS) attack on the node, or if it does not receive the packets due to collisions[†], poor transmission quality in wireless links, etc. Depending on the signature matching techniques used an insider may also report a good packet as bad. Bloom filters [14] are examples of such techniques. Bloom filters consider a packet or packet fragments to be suspicious whenever a hash function of the packet or packet fragments match that of an attack signature. Only suspicious packets are analyzed more thoroughly. This technique can be implemented using hardware and therefore signatures can be matched at line speed. But, this technique also results in false positives. The false positive rate can sometimes be non-negligible, e.g., sometimes 10% of the good packets have been reported to be suspicious [14]. Now when an insider node is overloaded or does not have enough resources it may report suspicious packets as bad without conducting a thorough analysis. Otherwise, if the node chooses to completely analyze suspicious packets it may have to drop several incoming packets which are more likely to be good.

We assume that every IDS active insider considers a bad packet to be good with probability p and a good packet to be bad with probability q. We focus on the efficacy of detection schemes given such a failure model. We do not consider the threat of compromise whereby insider nodes are under the control of the intruders. We assume that the values of p and q are known. This knowledge can be attained through online measurements. Due to space constraints, we do not describe such measurement techniques. Finally, we assume that the p and q are the same for every insider. Future research will be directed towards generalizing the framework for different values of p and q for different insiders.

There exist different strategies for selecting IDS active nodes. An obvious IDS placement strategy (host intrusion detection or HID) [15] is to execute the IDS at only the destinations of the sessions. Here, a node executes the IDS at its application layer, and can therefore analyze only the packets it receives as destination, and not those it relays. Another strategy is to use the network intrusion detection (NID) technique [15] where the IDS is executed on some selected insider nodes, which may be relays or end-hosts. Here, a node executes the IDS at its network layer, and can therefore analyze both the packets it relays and receives as destination. We consider NID in the rest of this paper.

The challenge in deploying NID is to appropriately select the IDS active nodes. A straightforward strategy is to execute the IDS on every insider node. Thus more bad packets will be examined. But, this clearly consumes significant energy and requires substantial computation. On the other hand, if the IDS are executed in very few nodes, then the resource consumption decreases but some bad packets may escape inspection. The security risk in each case will depend on the nature of the imperfections. In the first case, the detection rate will be higher as each packet is examined by higher number of

[†] Some collisions happen only due to promiscuous operation. Consider an ad hoc network with 3 insider nodes A, B, C. All nodes are IDS capable. Let both A and C be B's neighbors. But, A and C are not each others neighbors. Let B execute the IDS. If A and C simultaneously relay bad packets to outsider nodes, the packets collide at B. Since however the intruders transmit bad packets only rarely, such collisions of bad packets are rare.

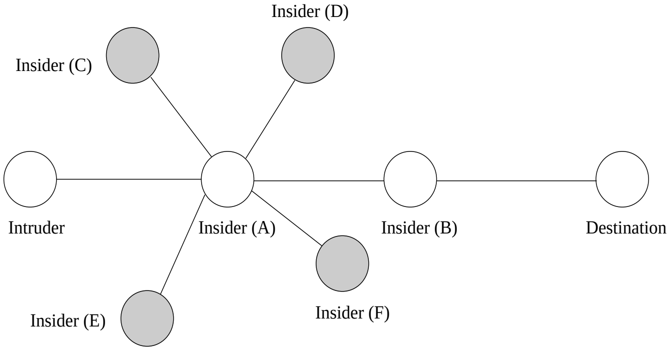


Fig. 1. This figure illustrates the coverage redundancy of an insider. The intruder attacks the destination. Insider nodes A and B relay the insider's packets to the destination. Node A is covered by IDS active insiders (C, D, E, F). When A relays a packet, C, D, E, F receive the packet in promiscuous mode. If any of these detect the packet to be bad, it reports its diagnosis to A. Based on the reports, A determines whether the packet is bad.

insiders. But, since more insiders examine each packet, more false positives may be generated and the number of good packets dropped may be higher (if a node drops packets marked as bad). The challenge now is to select the IDS active nodes such that even with these imperfections, the required detection and drop rates are achieved while limiting the resource consumption.

We assume that either a packet is not encrypted or only the transport layer payload of a packet is encrypted. This is the case with several protocols like PGP, SRTP, HTTPS etc. Then, IDS modules can detect attacks at the transport and lower layers, e.g., ping-of-death, TCP SYN flood, etc. without any knowledge of the encryption schemes. We do not consider link layer and network layer encryption protocols like IPSec since these protocols are typically used in enterprise environments which is beyond the scope of this paper. This is a standard assumption in several papers that investigate NID [15] .

3 Algorithms for Robust Intrusion Detection

We first consider the detection goals. Clearly, a detection goal would be to maximize the probability of detecting bad packets. But, since an IDS active insider reports some good packets as bad, we also need to minimize the probability of reporting a good packet as bad (*"false positive"*). It may not be possible to attain both goals simultaneously. This happens when increasing the detection rate involves increasing the false positive rate. Thus, our goal is to select the IDS active insiders among the IDS capable insiders so as to minimize the resource consumption subject to attaining a *risk* which is below an acceptable value. We quantify the risk as a weighted sum of the probability P_M that a bad packet is not detected (*missed detection*) and the probability P_F of false positive, $y_M P_M + y_F P_F$, where y_F, y_M are pre-specified weights. The resource consumption is the sum of the weights of all IDS active insiders.

The risk needs to be maintained below an acceptable value without any knowledge of the intruders, targets and the paths between them. This can be done only when the IDS active insiders are selected so that every packet is analyzed and the results of the analysis intelligently combined to reduce the errors in the decisions. If every IDS active insider could decide without any error ($p = q = 0$), then most of the packets would be analyzed if the IDS active insiders are selected so that every insider is a neighbor of at least one IDS active insider. In this case, every packet transmitted by an insider would be analyzed at least once and every insider detects whether a packet is bad without any error. Only the packets transmitted directly from an intruder to its target may not be analyzed, but the percentage of such packets is small [9]. This suggests that when insiders may decide erroneously, the IDS active insiders must be selected so that every insider is a neighbor of at least k IDS active insiders where $k > 1$. Now, the coverage redundancy may allow insiders to correct errors in decisions (Figure 1) [‡].

There are two questions that we now need to answer: (a) how does an insider optimally decide whether a packet is bad and (b) what is the optimal value of k. The issues clearly depend on each other. First consider the difficulties in determining (a). Due to the coverage redundancy several insiders may analyze a packet, and they may decide differently whether the packet is bad. The different decisions must be combined to determine whether the packet is indeed bad. For example, in Figure 1, if C and D detect a packet relayed by A to be bad and E and F determine otherwise, A needs to decide whether it should report an attack. The challenge is to aggregate the neighbors' decisions so as to minimize the expected risk. We attain this objective by developing an optimal aggregation scheme based on hypothesis testing framework. Now consider the difficulties in determining (b). We prove that under the optimal aggregation scheme the expected risk decreases with increase in k. But, the resource consumption also increases with increase in k. The challenge therefore is to select the minimum k that attains a tolerable risk when each insider optimally aggregates its neighbors' decisions. Once k is determined, we need to select the IDS active insiders so as to minimize the resource consumption or the total weight of the IDS active insiders subject to ensuring that every insider is covered by at least k IDS active insiders.

We determine the optimal k, the algorithms for aggregating the decisions and the optimal selection of IDS active insiders under the assumption that only one insider relays each packet. Using simulations, we investigate the performance of different algorithms when each packet is relayed by an arbitrary number of insiders.

We first obtain the optimal aggregation scheme at an insider i which has k IDS active neighbors. Consider a packet A selected with uniform probability among all packets relayed by i. Then, A is a good packet w.p. π_G, where π_G is the probability that an arbitrary packet is good [§] Now, the expected conditional risk for A is $H_i(k) = y_M P_{iM} + y_F P_{iF}$, where P_{iM} and P_{iF} are the respective probabilities of missed detection and false positive at i, which depend on i's aggregation scheme. Now, given k, i's optimal aggregation scheme is one that minimizes $H_i(k)$, and the minimum value of $H_i(k)$ is referred to as $H(k)$.

Theorem 1. *Given k, an insider attains $H(k)$ if it uses the following aggregation scheme. Let threshold $T = \left\lceil \frac{\ln \frac{y_F \pi_G}{y_M(1-\pi_G)} + k \ln \frac{1-q}{p}}{\ln((1-p)(1-q)/pq)} \right\rceil$. When $p + q < 1$,[¶] the insider node decides that a packet is bad if*

[‡] Using Figure 1, we describe a preliminary recovery protocol. If based on its neighbors' inputs, A determines a packet it relays to be bad, it reports its diagnosis to the next relay B. B holds each packet for some time before relaying it to the destination. If A reports the packet to be bad, B drops it. If A does not receive any report from B in the pre-determined interval, it delivers the packet to the destination. Details regarding the recovery protocol is beyond the scope of this paper.

[§] We assume that each insider knows π_G. Statistical techniques, e.g., Baye's minimax framework [16] can be used to develop optimal aggregation rules when this is not the case. Again, we can not describe such techniques due to lack of space.

[¶] Note that p and q are probabilities associated with different packets. Thus, $p + q$ can exceed 1.

and only if T or more of its IDS active neighbors inform that the packet is bad. When $p+q \geq 1,$ the insider node decides that a packet is bad if and only if fewer than T of its IDS active neighbors inform that the packet is bad.

Theorem 2. *When $p+q < 1$, $H(k) = y_F \pi_G \sum_{i=T}^{k} \binom{k}{i} q^i (1-q)^{k-i} + y_M(1-\pi_G) \sum_{i=0}^{T-1} \binom{k}{i} p^{k-i}(1-p)^i$. When $p+q \geq 1$, $H(k) = y_F \pi_G \sum_{i=0}^{T-1} \binom{k}{i} q^i (1-q)^{k-i} + y_M(1-\pi_G) \sum_{i=T}^{k} \binom{k}{i} p^{k-i}(1-p)^i$.*

We now present the intuition behind the results.

When $p+q < 1$, the probability of error is small. So, a large number of insiders are likely to report a packet as bad only when the packet is bad. Thus, an insider decides the packet is bad only when many of its IDS active neighbors report it as bad. When $p+q \geq 1$, probability of error is high. So, if a packet is bad, many insiders would report it as good. Thus, the previous policy is reversed. We computed the aggregation thresholds and the minimum risk $H(k)$ for an IDS active insider using the theory of hypothesis testing [11],[16]. Finally, note that the aggregation is optimum for an insider irrespective of whether it is IDS active. Since each insider is also its own neighbor, an IDS active insider executes the above aggregation rule considering both its and its other neighbors' analysis of each packet.

We now obtain the optimum value of k and the optimum set of IDS active insiders, D. First, we obtain the following result.

Corollary 1. *As k increases, $H(k)$ decreases.*

Intuitively, an insider can make better decisions, if it has more information, i.e., if it hears from more neighbors.

Let the tolerable accepted risk be γ. Since only one insider relays each packet, each packet has an expected risk of $H(k)$. Let $k_{\min} = \arg\min_k \{H(k) \leq \gamma\}$. The detection goal of attaining the tolerable expected risk subject to minimizing the total weight of the IDS active insiders is now satisfied if the IDS active insiders are selected so as to minimize their total weight subject to ensuring that each insider has at least $k_{\min}$ IDS active neighbors. We now discuss how to select the IDS active insiders so as to attain the above goal. We first introduce some terminologies.

Definition 2. *A k-multicover in G is a set of insiders such that every insider in G is covered by k insiders in the set.*

Definition 3. *An IDS capable k-multicover is a k-multicover such that its members are IDS capable.*

Definition 4. *An IDS capable minimum weighted k-multicover is an IDS capable k-multicover with minimum total weight among all IDS capable k-multicovers.*

Clearly, the set of IDS active insiders need to be an IDS capable minimum weighted k-multicover in G. It is well-known that computing a minimum weighted k-multicover is an NP-hard problem [17]. This motivates the following lemma.

Lemma 1. *Optimally selecting the IDS active insiders is an NP-hard problem.*

There may not be any k-multicover in G, e.g., when an insider node in G is covered by at most $k-1$ insider nodes. In this case, we have to opt for maximum possible coverage, which we explain later.

We next present an approximation algorithm (APPROX) for selecting the IDS active insiders. This has been obtained by modifying a greedy algorithm in [18] to accommodate IDS incapable insiders and still provide performance guarantees.

We introduce some terminologies required to describe the algorithm. The algorithm progressively adds IDS active insiders in a set D, which is initially empty. An insider in G is "satisfied" if it is covered by k or more insiders in D, and unsatisfied otherwise. Initially, every insider is unsatisfied. Let $US(u)$ be the number of unsatisfied neighbors of an insider node u in G. Let I be the set of IDS capable insiders in G. When the algorithm terminates, if there exists at least one IDS capable k-multicover in G, all insiders are satisfied and D is a k-multicover.

1. Let every insider be unsatisfied and $D = \phi$.
2. Let u be an IDS capable insider such that $w_u/US(u) = \min_{v \in I \setminus D} w_v/US(v)$.
3. $D = D \cup \{u\}$.
4. If any of u's neighbors v has k neighbors in D, v's status changes to satisfied.
5. Terminate if all the neighbors of the insiders in $I \setminus D$ are satisfied. Otherwise, go to step 2.

Note that if an insider u is not added in D, then insiders must be added to cover $US(u)$ insiders. Thus, $US(u)$ reflects the reduction in the coverage requirement brought about by adding u in D. The algorithm requires $|I|$ or less iterations, and has complexity $O(|I|)$.

Theorem 3. *Let G have at least one IDS capable k-multicover. Then, D is an IDS capable k-multicover with weight (resource consumption) at most $\sum_{i=1}^{d}(1/i)$ times that of the weight of an IDS capable minimum weight k-multicover in G, where d is the maximum degree of an insider node in G.*

Theorem 4. *Let G have no IDS capable k-multicover. An insider either has at least k IDS active neighbors or all its IDS capable neighbors are IDS active.*

It is in this sense that D maximizes coverage in G, when G has no IDS capable k-multicover. Each insider can decide whether to execute the IDS in a distributed manner, if it learns $\min_{v \in I \setminus D} w_v/US(v)$ by exchanging messages with its neighbors. This can be attained if a pre-determined root node broadcasts a packet with a value 0 and every IDS capable insider u which is IDS inactive and has $US(u) > 0$ updates the content of the message with the minimum of the current content and $w_u/US(u)$, and forwards it along the broadcast tree. The leaf nodes of the tree return the packet towards the root. If an insider executes the IDS, it informs its neighbors, and they change their status to satisfied if required. Each insider also informs its neighbors about whether it is satisfied. The insiders learn about the updated $\min_{v \in I \setminus D} w_v/US(v)$ by another broadcast. The root node does not broadcast when it receives a return packet with content 0. Clearly, at most $|I|$ broadcasts are necessary if the insiders do not move.

This algorithm for selecting the IDS active insiders is oblivious to the position of the outsiders, and is therefore not affected by their movements. But, the IDS active set must be recomputed each time an insider node's neighborhood changes due to its or its neighbors' movements. The computations and the related message exchanges consume significant resources particularly when they are executed frequently, i.e., when the insider nodes move rapidly. We now present computationally simple algorithms that do not require any re-computation with movement of either insider or outsider nodes, and require only limited message exchange when insider nodes move. The disadvantage is that we have not been able to prove any approximation bound for any of these algorithms. We evaluate them using simulation.

First we modify APPROX. Now, an IDS capable insider node u periodically exchanges messages with its neighbors to learn $w_v/US(v)$ for each neighbor v. After obtaining this information, u executes the IDS if and only if $US(u) > 0$ and $w_u/US(u)$ is the minimum in u's neighborhood. Then, u informs its neighbors about whether it is IDS active. This modified version is referred to as MUN (Maximum Unsatisfied Neighbors). Clearly, if the insider nodes do not move, the set of IDS active insiders stabilizes after some time. When insider nodes move, the decisions are updated in their neighborhoods, but the updates involve simple computations and limited message exchange. Since the decisions now depend on local instead of global minimums, we could not prove any approximation bound. Nevertheless, extensive simulations show that MUN and APPROX have similar performance.

Now, we consider a naive algorithm, Random Placement (RP), in which every IDS capable insider node executes the IDS with a probability which can be selected so as to regulate the resource consumed and the expected risk. For example, if this probability is high, then a large number of insiders are IDS active. Thus, the scheme consumes a lot of resource but the expected risk is likely to be low. This algorithm requires no message exchange. In Section 4, we compare the performances of APPROX, MUN and RP, and determine when each may be deployed.

4 Performance Evaluation

Using ns2-simulations, we compare the performance of the approximation algorithm, APPROX, and the computationally simple heuristics MUN and RP for selecting the IDS active insiders. We consider the optimal aggregation rule in each case. The simulations allow us to investigate the effect of the factors we did not consider in the analysis such as arbitrary number of hops between the intruder and the target, mobile insiders etc. We also evaluate the benefits of intelligently selecting the IDS active nodes, and accordingly decide the appropriate algorithm for any desired tradeoff between risks and resource consumption. Sample computations suggest that the approximate selection algorithm, APPROX, closely approximates the optimal solution, and the performance difference is generally much less than the upper bound presented in Theorem 3; we do not include these comparisons due to space constraint [11].

We consider networks with different types of node mobility, e.g., mobile intruders and static insiders, mobile intruders and mobile insider nodes etc. Each mobile node moves as per the random way point model with a maximum speed of 20 m/s and pause time 10 sec. For each combination, we measure averages over 300 different topologies. Each topology consists of a single intruder, a single target and 100 insider nodes. The insider nodes are uniformly distributed in a square of side 600m. Every insider is IDS capable and has unit weight. Every node has transmission radius 150m. Each attack consists of a single packet, and 20% of the packets transmitted by the intruder are bad (i.e., constitute attacks). Thus $\pi_G = 0.8$. We assume $y_M = y_F = 1$. For each topology, we measure the expected risk as the sum of the bad packets that are not detected and the good packets that are reported as bad divided by the total number of packets. We measure the detection cost as the total number of IDS active insiders.

In figure 2(a) and (b) we compare the analytical results with the simulation results. We plot $H(k)$ (obtained from Theorem 2) and the expected risk measured in the simulations as a function of k. We consider networks where every packet is relayed by a single insider node. For each $k,$ in each topology, we select the IDS active insiders using the APPROX algorithm. Then, we select an insider uniformly among all the insiders and measure the risk when it relays the packets. The intruder and its target are selected within the transmission range of this insider. Here, we assume that all nodes are static. We consider different values of p, q: (i) two sets of p, q with $p + q < 1$ (figure 2(a)), and ii) two sets of p, q with $p + q > 1$ (figure 2(b)). Since the number of insiders is $100,$ in most trials each insider has fewer than 9 neighbors. Thus, we could vary k only until 8. We see that the expected risk measured in simulations is close to, but lower than $H(k)$. The difference can be explained as follows. APPROX only ensures that every insider has at least k IDS active neighbors. Thus, some insiders have more than k IDS active insiders. But, $H(k)$ is the minimum expected risk at an insider with exactly k IDS active insiders. Since the expected risk decreases with increase in k, the expected risk measured in the simulations is less than $H(k)$.

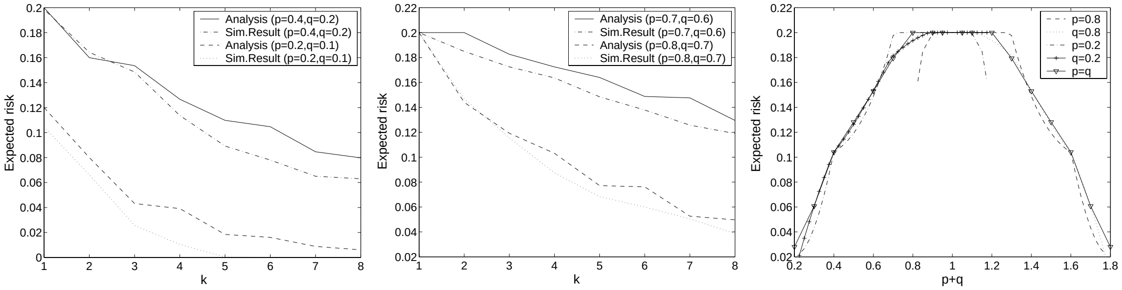
In figure 2(c), we plot $H(k)$ as a function of $p+q$ for different values of p and q. We select $k = 3$ here. Note that $H(k)$ is small when $p + q$ is either small or large, and maximum around $p + q = 1$. The plots are symmetric around $p + q = 1$. This is because the aggregation process is most prone to error around $p + q = 1$. When $p + q$ is small (i.e., $p + q << 1$), only few insiders will erroneously determine whether a packet is bad. Thus, if a packet is bad, most insiders will report it as bad, and if a packet is good, most insiders will determine it as good. For $p + q < 1$, the optimum aggregation rule considers a packet as bad if and only if more than a certain number of insiders report it as bad. Thus, the aggregation process is less likely to have errors and hence $H(k)$ is small. Now, when $p+q$ is large (i.e., $p + q >> 1$), a large number of insiders will erroneously determine whether a packet is bad. Thus, if a packet is bad, only few insiders will report it as bad, and if a packet is good, again only few insiders will determine it as good. For $p + q > 1$, the optimum aggregation rule considers a packet as bad if and only if fewer than a certain number of insiders report it as bad. Thus, the aggregation process is again less likely to have errors and hence $H(k)$ is small. When $p + q \approx 1$, for each packet, the number of insiders reporting it as bad and good are similar. Thus, the reports have less correlation with the packet's nature. Hence, the aggregation process is prone to more errors and $H(k)$ is larger.

Using simulations, we now compare the efficacy of different algorithms for selecting the IDS active insiders (Figure 3). We consider $p = q = 0.05$. We plot the ratio of the expected detection costs of different algorithms (RP and MUN, and MUN and APPROX) as a function of the expected risk. For each risk value γ and algorithm, we determine the lowest value of k that attains expected risk less than γ, and determine the expected detection costs of the algorithm at this value of k. We consider both static and mobile insider nodes. The intruder and its target are now selected uniformly. Thus the path between them, which is selected by AODV, consists of arbitrary number of hops.

We first consider static insiders (Figure 3(a)). First, note that MUN and APPROX perform similarly all through. Thus, intelligent selection based on a local criteria can perform similar to that based on global criteria in this problem. We now assess MUN's advantage over the naive RP. For intermediate values of acceptable risk, MUN reduces the detection cost of RP by about half. In this region, the risk values require intermediate values of k, and MUN's intelligent selection of IDS active nodes attains the same coverage as RP while using half the number of IDS active insiders as RP. For very low expected risks, RP and MUN have similar detection costs. This is because k needs to be large in this case, and equals the number of neighbors of the insiders in many cases. But then both RP and MUN would need to execute IDS in a large fraction of insider nodes, e.g., around $70-80\%$. Thus both have similar detection costs. Similarly, when the tolerable expected risk is high, both RP and MUN need very few IDS active insiders. Thus, again both have similar detection costs. In both extremes, intelligent coverage algorithms do not provide significant benefits. The trends and conclusions remain similar for mobile insiders (Figure 3(b)). Given that APPROX requires global recomputation every time an insider moves, APPROX can not be used in this case. We have investigated the performance of these strategies in many other scenarios. The results are similar for other transmission radii, mobility parameters, number of insider and intruder nodes, p, q, etc. [11]. We do not show these figures due to lack of space.

Summarizing, MUN's performance is similar to APPROX that has analytical performance guarantees and MUN's performance is significantly better than that of RP. MUN is much simpler than

APPROX, but is little more complex than RP. Thus, MUN attains a good balance between complexity and performance for providing robust intrusion detection in ad hoc networks.



(a) p+q < 1

(b) p+q > 1

(c) k=3

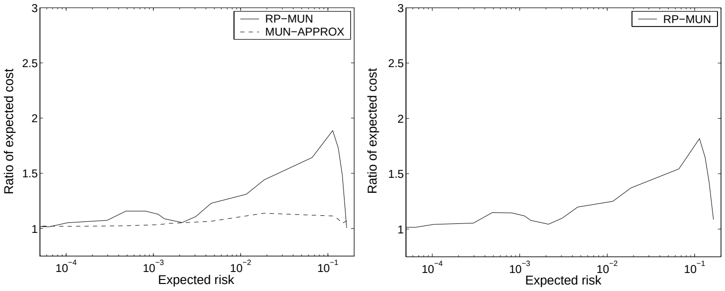
Fig. 2. In figures (a) and (b) we plot the expected risk as a function of k when $p+q < 1$ and $p+q > 1$ respectively. In figure (c), we plot the expected risk as a function of $p + q$ when k=3.

5 Conclusion

We consider ad hoc networks with imperfections in the nodes performing intrusion detection tasks. Combining tools from the theories of hypothesis testing and approximation algorithms, we develop a framework to counter different threats while minimizing the resource consumption. We obtain computationally simple optimal rules for aggregating and thereby minimizing the errors in the decisions of the nodes detecting the intrusion. We also show that the optimal selection of nodes that execute the detection modules is an NP-hard problem. Hence, we present a polynomial complexity selection algorithm APPROX that attains a guaranteeable approximation bound. This algorithm is modified to allow for seamless operation in time varying topologies. The modified version (MUN) needs only simple computations and local message exchanges when nodes move. Nevertheless, our simulation reveals that MUN's performance is similar to that of APPROX. In many cases MUN reduces the detection cost by about half as compared to a naive heuristic. We conclude that MUN provides a good balance between complexity and performance for attaining robust intrusion detection in ad hoc networks.

References

1. Cole, E.: Hackers Beware. New Riding Publishing (2001)
2. Cheswic, W., Bellovin, W.: Firewalls and Internet Security. Addison Wesley (1999)
3. Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., Paxson, V.: Stream control transmission protocol. RFC 2960 (2000)
4. Nikita Borisov, Ian Goldberg, D.W.: Intercepting mobile communications: The insecurity of 802.11. In: Proceedings of MOBICOM 2001. (2001)



(a) Static insiders

(b) Mobile insiders

Fig. 3. We plot the ratio of the expected cost of different algorithms as a function of the expected risk. Here, $p = q = 0.05$. We compare RP and MUN and MUN and APPROX for static insiders in figure (a). We compare RP and MUN for mobile insiders in figure (b).

5. Denning, D.: An intrusion detection model. In: IEEE Transactions on Software Engineering. Volume SE-13. (2001) 222–232
6. Garfinkel, S., Spafford, G.: Practical UNIX and Internet Security. 2nd edn. O'Reilly and Associates (1996)
7. Ko, C., Brutch, P., Rowe, J., Tsafnat, G., Levitt, K.: System health and intrusion monitoring using a hierarchy of constraints. In: 4th International Symposium, Recent Advances in Intrusion Detection. (2001) 190–204
8. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. In: Mobile Computing and Networking. (2000)
9. Subhadrabandhu, D., Sarkar, S., Anjum, F.: Efficacy of misuse detection in adhoc networks. In: Proceedings of the IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON). (2004)
10. Zhang, Y., Lee, W.: Intrusion detection in wireless ad-hoc networks. In: Proceedings of the Sixth International Conference on Mobile Computing and Networking (MobiCom). (2000)
11. Subhadrabandhu, D., Sarkar, S., Anjum, F.: Misuse detection with imperfect defenders in adhoc networks. Technical report, University of Pennsylvania Technical Report, http://www.seas.upenn.edu/∼swati/publication.htm (2004)
12. Ramanujan, R., Kudige, S., Nguyen, T., Takkella, S., Adelstein, F.: Intrusion-resistant ad hoc wireless networks. In: Proceedings of MILCOM. (2002)
13. Ptacek, T., Newsham, T.: Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, an SNI Technical Report, http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps (1998)
14. Dharmapurikar, S., Krishnamurthy, P., Sproull, T., Lockwood, J.: Deep packet inspection using parallel bloom filters. In: Micro, IEEE. Volume 24. (2004) 52–61
15. McHugh, J.: Intrusion and intrusion detection. International Journal of Information Security (2001)
16. Poor, H.V.: An Introduction to Signal Detection and Estimation. 2nd edn. Springer-Verlag (1994)
17. Garey, M.R., Johnson, D.S.: Computers and Intractability. W. H. Freeman and Company (2000)
18. Vazirani, V.: Approximation Algorithms. Springer (2001)