Enhanced Intrusion Detection System for Discovering Malicious Nodes in Mobile Ad hoc Networks

Nidal Nasser and Yunfeng Chen

Department of Computing & Information Science
University of Guelph
Guelph, ON, Canada N1G 2W1
{nasser, yunfeng}@cis.uoguelph.ca

Abstract – As mobile wireless ad hoc networks have different characteristics from wired networks and even from standard wireless networks, there are new challenges related to security issues that need to be addressed. Many intrusion detection systems have been proposed and most of them are tightly related to routing protocols, such as Watchdog/Pathrater and Routeguard. These solutions include two parts: intrusion detection (Watchdog) and response (Pathrater and Routeguard). Watchdog resides in each node and is based on overhearing. Through overhearing, each node can detect the malicious action of its neighbors and report other nodes. However, if the node that is overhearing and reporting itself is malicious, then it can cause serious impact on network performance. In this paper, we overcome the weakness of Watchdog and introduce our intrusion detection system called ExWatchdog. The main feature of the proposed system is its ability to discover malicious nodes which can partition the network by falsely reporting other nodes as misbehaving and then proceeds to protect the network. Simulation results show that our system decrease the overhead greatly, though it doe not increase the throughput obviously.

I. INTRODUCTION

A Mobile Ad hoc Network, or MANET, consists of a group of cooperating wireless mobile hosts (nodes) that dynamically constructs a short lived and self-configuring network without the support of a centralized network infrastructure. The mobile nodes can be cell-phones, PDAs and laptops and typically support several forms of wireless connectivity like 802.11, IrDA, Bluetooth, etc.

One advantage of wireless networks is the ability to transmit data among users in a common area while remaining mobile. However, the range of transmitters or their proximity to the wireless central points limits the distance between peers. Mobile Ad hoc networks mitigate this problem by allowing out of range nodes to route data through intermediate nodes, i.e., each send its own data as well as routes and forwards data on behalf of other nodes.

The MANET may operate in a standalone fashion, or may be connected to the larger Internet. Minimal configuration and quick deployment make ad hoc networks suitable for using in emergency circumstances where an infrastructure is unavailable or unfeasible to be installed like natural or human-induced disasters, military conflicts and medical emergency situations.

There have been significant improvements related to the different characteristics and topics in MANET, particularly in the fields related to routing protocols, clustering protocols, location and mobility prediction. However, the security aspects of MANET have been rarely addressed. The security goals of MANET include availability, integrity, authentication, confidentiality and non-repudiation [9].

One of the primary concerns related to ad hoc networks is to provide a secure communication among mobile nodes in a hostile environment. The nature of mobile ad hoc networks poses a range of challenges to the security design. These include an open decentralized peer-to-peer architecture, a shared wireless medium and a highly dynamic topology. This last point is where the main problem for MANET security resides: the ad hoc networks can be reached very easily by users, but also by malicious attackers. If a malicious attacker reaches the network, the attacker can easily exploit or possibly even disable the mobile ad hoc network. Conventional methods of identification and authentication are not available since the availability of a Certificate Authority or a key Distribution Center cannot be assumed.

In this paper, we propose an intrusion detection system ExWatchdog by extending Watchdog proposed in [3]. Our solution is particularly aims at solving weaknesses presented by Watchdog: a malicious node falsely reports other nodes as misbehaving while in fact it is the true culprit. Each node maintains a table that records the number of packets the node sends, forwards or receives respectively. When receives a report about misbehaving nodes, the source of a communication can send a message to the destination to check if the sums of packets the two parts stores are equal. If they are equal, then the real malicious node is the node that reports others nodes as misbehaving. Otherwise, nodes being reported malicious do misbehave.

The remainder of the paper is organized as follows. Section II presents the background review and related work that are important for the understanding of the material to follow. Section III introduces our intrusion detection and response system, ExWatchdog. Section IV reports the
simulation results and discussion. Finally, conclusions drawn from the paper and future work are given in Section V.

II. BACKGROUND AND RELATED WORK

In this section, we present an overview of significant concepts that are important for the understanding of the material to follow. We first briefly introduce the routing protocols in MANET. Then we describe the intrusion detection and response systems in MANET and how they differ from systems in traditional systems like the Internet. Finally, we give a quick review of the proposed intrusion detection and response systems Watchdog, Pathrater and Routeguard. Our system ExWatchdog is an extension of Watchdog, i.e., ExWatchdog is an intrusion detection system, which needs to cooperate with response system like Pathrater or Routeguard to protect the system from potential intrusion.

A. Routing Protocols in Ad hoc Networks

In mobile ad hoc networks, due to the limited wireless transmission range, it is usually the case that paths between source nodes and destination nodes require multiple hops. In this case, every node may act as a router to forward packets from the source to the destination. Also, due to the nodes mobility, it is necessary to change the existing routes in order to maintain the connection between them. A MANET routing protocol must be able to quickly detect and respond to such state changes in order to minimize degradation in services provided to existing sessions.

There are many different routing protocols that can be used in MANET. In general, they can be classified into two main groups: proactive routing and reactive routing, depending on the time when route are determined. Proactive routing constantly formulates routing choices in order to have accessible paths available for nodes that need to send packets [6]. Reactive routing on the other hand only finds a route when requested; as soon as a node has a packet to transmit, it queries the network for a route [2, 4, 5, 7, 8].

Dynamic Source Routing Protocol (DSR) [2] is an on-demand, source routing protocol. On-demand means that route paths are discovered at the time a source sends a packet to a destination for which the source has no path. There are two major phases in the operation of DSR: Route Discovery and Route Maintenance. When a node in the ad hoc network attempts to send a data packet, it first checks its cache table to look for a route to the destination. If the route in cache is expired or if it has no route, a route discovery process is launched to discover one. The second main function of DSR is route maintenance. If any link on a route is broken, the source node is notified by an intermediate node that detects the link break when forwarding a packet to the next node in the route path. The source node removes any route using this link from its cache, or removes the portion of the route that uses that broken link. The source must also try another path or do a new route discovery if it does not have another path.

B. Intrusion Detection and Response Systems

The basic preconditions for intrusion detection are that there are intrinsic and observable characteristics of normal behavior that can be collected and analyzed and that it is possible to use those characteristics and behaviors to distinguish normal from abnormal behavior.

Most traditional intrusion detection systems take either a network-based or a host-based approach to recognizing and deflecting attacks. Network-based IDS listen on the network, to capture and examine individual packets flowing through a network. Host-based systems are concerned with what is happening on each individual host.

However, due to some specific features of MANET, neither host-based nor network-based IDS is suitable for MANET. Thus, the IDS for MANET have been proposed to work in a collaborative way and as part of the existent routing protocols.

In MANET, intrusion detection and response systems should be both distributed and cooperative in order to fulfill the needs of mobile ad hoc networks. In the architecture proposed in [10], every node in the mobile ad hoc network participates in intrusion detection and response. Since every node cannot trust its neighboring nodes, it is responsible for detecting signs of intrusion locally and independently. However, neighboring nodes can collaboratively exchange messages in case of a suspicious situation or confirmed intrusion detection.

C. Related Works

1) Watchdog And Pathrater: Marti et al. [3] discussed two techniques (Watchdog and Pathrater) that improve throughput in mobile ad hoc networks in the presence of compromised or malfunctioning nodes that consent to forward packets but are unsuccessful in doing so. A significant increase in throughput was observed in mobile ad hoc networks [2] by complementing the DSR protocol with the Watchdog/Pathrater scheme.

Watchdog relies upon DSR and each node takes part in the intrusion detection and response by surveillance of its downstream node, on the route from source to destination, with the purpose of making sure that it has retransmitted the packet without alteration. If this downstream node does not forward the packet, it is misbehaving. To mitigate the effects of a misbehaving node, Marti et al. introduce Pathrater, which chooses a path from source to destination based upon a simple rating algorithm, instead of the shortest path. Pathrater is run by each and every node in the network.

Watchdog technique has advantages and weaknesses that are discussed in [3]. DSR with the Watchdog has the advantage that it can detect misbehavior at the forwarding level and not just the link level. Watchdog’s weaknesses are...
that it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion, and 6) partial dropping. Hasswa et al. [1] also discuss the weaknesses of Pathrater. The major weaknesses, related to the rating scheme, include: 1) inflexible binary states, 2) behavioral deceit, 3) new node anonymity, 4) re-entrance of previously malicious node, and 5) encouraging selfishness and greed.

2) Routeguard [1]: Similar to the Pathrater, Routeguard is run by each node in the network. Each node stores a rating for all the nodes it knows. However, as an improvement to Pathrater, Routeguard assigns ratings to nodes and calculates a path metric in a refined way. Routeguard introduces a more detail and natural classification system that rates each node in the network into one of the five classes: Fresh, Member, Unstable, Suspect, or Malicious. Each node is treated differently depending on its status and rating.

III. ExWatchdog Intrusion Detection System

In this section, we describe the intrusion detection and response system named ExWatchdog. Just as its name implies, ExWatchdog is an extension of Watchdog and its function is also detecting intrusion from malicious nodes and reports this information to the response system, i.e., Pathrater or Routeguard.

A. Malicious Nodes in Watchdog: Problems, Discussion and Motivation

Either in Watchdog or Routeguard, each node updates ratings of nodes it knows according to the information provided by any node in the network.

![Figure 1: Malicious node A falsely report B as misbehaving in order to partition the network](image)

One serious problem possibly happens when a node falsely report, whatever maliciously or accidentally, other nodes as misbehaving. A malicious node could partition the network by claiming that some nodes following it in the path are misbehaving. For instance, in Figure 1, node S is the source A is malicious, B is the destination in a communication. Node A could report that node B is not forwarding packets however in fact it is. This will cause S to mark B as misbehaving when A is the real culprit.

This behavior, as the authors of Watchdog explains in [3], can be detected like this. “Since A is passing messages on to B (as verified by S), then any acknowledgements from D to S will go through A to S, and S will wonder why it receives replies from D when supposedly B dropped packets in the forward direction. In addition, if A drops acknowledgements to hide them from S, then node B will detect this misbehavior and will report it to D.”

The authors take it for granted that Watchdog could detect such problem. However, if the intention of the malicious node A is to partition the network, then Watchdog is helpless. Thinking further, we can find that the problem occurs just when S and A mark B as misbehaving and D and B mark A as misbehaving. If node B detects misbehavior of A and reports it to D, then D and B will regard A as malicious. At the same time, S and A regard B as malicious because A report to S that B is malicious. Thus in some sense, the consequence is that the network is divided into two parts: S, A and B, D separately.

Figure 1 shows such a condition. A tells S that B is misbehaving then S and A mark B as malicious. B also tells D that A is misbehaving then D and B mark A as malicious. Thus the link between A and B is cut off and can not be used any more.

![Figure 2: Malicious node A falsely reports all nodes on the path from source to destination as misbehaving in order to partition the network](image)

This situation of network partition is especially serious when intermediate node A reports all the nodes on the routing path from source S to destination D being malicious. We consider the case when node A is on all the paths from S to D and A is a malicious node. Figure 2 shows the result when malicious node A reports all nodes, F and B, on the path from the source S to the destination D being malicious.

In Figure 2, there are two paths from S to D after Route Discovery:

- S -> A -> B -> D, and
- S -> A -> F -> D.

If A reports B and F as misbehaving successively then S marks B and F as malicious. As a consequence, D, B and F make A as malicious because A drops acknowledgements to S. If there are other nodes that communicate with D through S, S2 for example, then S will notify these nodes and they
will also mark B and F as malicious. And the network will be partitioned just as by the dashed line in Figure 2. This case may be the worst condition. If there exists a path from S to D and A is not on that path (S -> C -> E -> G -> D), then S still can communicate with D. But even in that case, the malicious node A partly gains its end:

1) S will regard node B and F as malicious and will not forward packets for them.
2) S has to select a suboptimal path from its route table or launch a new Route Discovery, which causes extra overhead.
3) When some other nodes, such as S2, launch Route Discovery to node D and node S happens to be an intermediate node, S will not return a Route Reply with node B and F in the route path, which might be the optimal path. Or if S2 already has routing paths to D and paths S -> A -> B -> D, and S -> A -> F -> D are part of them, then S will notify S2 of B and F being malicious and S2 may also select a suboptimal path from its route table or launch a new Route Discovery.

Another possibility is that using the path D->F->E->C->S, D sends a message to S notifying that A is malicious and S should modify its routing table. If this happens, S will get more confused about which node tells the truth and which node it should trust.

This problem is also applied to Routeguard because Routeguard only refines the way of ratings, lacking the mechanism to check the correctness of information reported from nodes acting as Watchdog.

**B. ExWatchdog System Description**

Our system works as an extension to the Watchdog and aims to detect nodes that falsely report other nodes as misbehaving. In general, these nodes are malicious instead of selfish. They can cause more serious damage to the network performance. The proposed system still uses the Routeguard as the response system.

In our system, we make the following assumption: Some encryption mechanisms are used and the key lengths are sufficiently long, making it infeasible to compute or guess a private key knowing only the public key, but on the other hand do not make the computation expensive for the mobile devices. Under this assumption, malicious nodes cannot tamper packets.

We implement the intrusion detection function by maintaining a table that stores entry <source, destination, sum, path>. Whatever the current node is, the source, the destination or the intermediate node, it inserts such an entry into the table when sending, forwarding or receiving packets for the first time. The value of each field is:

*Source*: the address of source.

*Destination*: the address of destination.

*sum*: the total number of packets that the current node sends, forwards, or receives using the route path *Path* as source, intermediate node or destination respectively.

*path*: the route that is used for the communication between *source, destination*. The *path* is a list of nodes addresses or an path ID for simplicity.

When an intermediate node on a route path reports to the source that its next hop is malicious, the source will not immediately decrease the rating of the malicious node. Instead, it will send a message to the destination using an alternative path in the route table. The message contains <source, destination, sum, malicious_node_address>. *Source*, destination and sum are the same as the above. malicious_node_address is the address of the node being reported malicious. The source node then searches a path that has no malicious node in it from the routing table. If there is not such a path available, the source then launch a Route Discovery to find a new one. After finding a path, the source sends the message using the found path.

Upon receiving the message, destination node will search its own table to see if there is a match. If there is not a matching entry in the table, it means the node is malicious and the destination node returns a message to the source confirming that the malicious node is really malicious. If there is, destination node then compares the sum field of the passing in message with the one found in the table. If the two sums equal, it means that the malicious node forwards all packets that the source sends thus it is not malicious. On the contrary, if the two sums are not equal, the node falsely report might be malicious. Then Routeguard will use this information to update the rating of corresponding node.

Compared to Watchdog, our solution has the same advantages. At the same time, it solves one big weakness: false misbehavior. It can detect if nodes falsely report other nodes as misbehaving. As stated earlier, false reporting may result in network partition and further decrease network performance.

However, there is limitation in our solution. If the real malicious node is on all paths from specific source and destination, then it is impossible for the source node to confirm with the destination of the correctness of the report. For this case, we do not and can not take any action because we do not know who lies and can not either check. The Routeguard decreases rating of neither the reporting node nor the reported node.

Figure 3 gives the pseudo code of maintaining the additional work for ExWatchdog.

**IV. PERFORMANCE EVALUATION**

The performance of ExWatchdog is evaluated in this section. The simulation model is described in Section IV.A and the results are presented and discussed in Section IV.B.

**A. Simulation Model**

A simulation model for ExWatchdog system has been developed in Network Simulator (NS-2) [9]. Our simulations take place in a network with 300m x 300m flat space filled with a scattering of 50 wireless mobile nodes.
The nodes communicate using 10 constant bit rate (CBR) node-to-node connections with a data rate of 4 packets per second. All nodes move in random mode with speed varying from 0 meter/second to 3 meter/second. The simulation time is 100 seconds. In all experiments, misbehaving nodes are nodes that agree to forward packets (they do not alter the content of packets) and then fail to do so due to overload, selfishness, maliciousness, or brokenness. In our simulations, the misbehaving nodes can damage the network performance especially by falsely reporting that other normal nodes as misbehaving. We put special attention on how this malicious action impact network performance. Of the 50 nodes in the simulated network, some variable percentages of the nodes misbehave. The percentage of misbehaving nodes varies from 0% to 40% in 10% increments. First we run the simulation with 30% misbehaving nodes which misbehave as falsely reporting. Then we repeat the experiments with 80% misbehaving nodes which misbehave as falsely reporting.

Nodes sending, forwarding, or receiving packets do:

```plaintext
//search if the entry exists in table
One_entry = search_entry();

if (One_entry) {
    // add new entry to table
    add_new_entry();
} else
    // update the sum of existed entry
    Update_sum();
```

The destination node verifies if the node is malicious does:

```plaintext
One_entry = search_entry();

if (One_entry) return false;
else {
    sum_in_table = One_entry.getSum();
    sum_from_source = msg_from_source.getSum();
    if (sum_in_table == sum_from_source)
        return true;
    else
        return false;
}
```

Figure 3: Abstract pseudo code of ExWatchdog operation

**B. Simulation Results**

The simulation results are presented in Figures 4-7. We use the same metrics with [3] to evaluate our extension: Throughput and Overhead. Throughput is the percentage of sent data packets actually received by the intended destinations. Overhead is the ratio of routing-related transmissions to data transmissions in a simulation.

In Figure 4 and Figure 5, we change the percentage of generally misbehaving nodes from 0% to the highest 40%, with the percentage of falsely reporting misbehaving nodes fixed at 30%.

![Figure 4: Overall Network Throughput As a Function of the Percentage of Misbehaving Nodes](image)

![Figure 5: Network Overhead As a Function of the Percentage of Misbehaving Nodes](image)

Figure 4 shows the total network throughput. In the case where the network contains no misbehaving nodes, the two curves achieve a throughput around 95%. In this case, we do not see the advantage of ExWatchdog due to the absence of the special misbehaving nodes, which is the goal of ExWatchdog. However, the results change after the 20% misbehaving node case.

In the worst case when 40% of the nodes are misbehaving, the average throughput of Watchdog degrades by 18% while the average throughput of ExWatchdog decreases by only 7%. Compared to Watchdog, ExWatchdog increases the throughput by up to 11%. The performance improvement is not too much.

In Watchdog, when a malicious node falsely reports to the source that the next hop is malicious, the source will select another path to send packets even it cannot detect the false reporting. The intention of specific malicious nodes is to partition the network instead of dropping packets. Thus it does not influence the number of data packets received by the destinations.

Figure 5 shows the overhead. Compared to Watchdog, ExWatchdog decreases the overhead by up to 35% when 40% nodes are misbehaving. The reason is that Watchdog...
will keep flooding Route Discovery if all the paths from the source to the destination are falsely reported as having one malicious node that actually is normal. This brings plenty of extra transmission of control packets. However, in ExWatchdog, the source will not select a new path or broadcast Route Discovery upon receiving the false report. It will check the correctness. If it is a false report, the source will only decrease the rating of the falsely reporting node and does not take other actions.

When 80% of the malicious nodes misbehave as falsely reporting, the performance difference is obvious. In Figure 6, the overall throughputs of both Watchdog and ExWatchdog do not decrease acutely. This is the same result as we explain before, i.e., some but not a great number of packets are used to discover new routes. Thus, the influence is trivial.

In Figure 7, however, for the communication overhead of Watchdog, there is a large augment when malicious nodes increase from 30% to 40%. Through the analysis of the trace file, we found when the 16 (50*40%*80 = 16) malicious nodes that falsely reports distribute evenly in the network, nodes in Watchdog have to broadcast Route Request once and again, which generates a large number of communication overhead. In ExWatchdog, the overhead of recognizing the special malicious node is constant, i.e., once these malicious nodes are found, no extra communication overhead will be generated.

V. CONCLUSION AND FUTURE WORK

Ad hoc networks are an increasingly promising area of research with lots of practical applications. However, MANETs are extremely vulnerable to attacks due to their dynamically changing topology, absence of conventional security infrastructures and open medium of communication, which, unlike their wired counterparts, cannot be secure.

In this paper, we present our intrusion detection system ExWatchdog, which is based on one proposed solution Watchdog. ExWatchdog solves a fatal problem of Watchdog, i.e., a malicious node can partition the network by falsely reporting other nodes as misbehaving.

We use Throughput and Overhead as metrics to evaluate the performance of ExWatchdog with some nodes being malicious nodes that falsely report other nodes as misbehaving. For each metric, we test Watchdog and our solution separately. The simulation results show that our solution decrease the overhead greatly, though it doe not increase the throughput obviously.

For future work, we’d like to research a more efficient and reliable method to check if the reporting node is the real culprit because our solution is based on the assumption that the malicious nodes cannot tamper the packet. This assumption may not practical in some application.

REFERENCES