Intrusion Detection in Wireless Ad Hoc Networks

Banshilal Patidar, Pinaki A.Ghosh
Department of Computer Science and Engineering
Bansal Institute of Science and Technology
Bhopal (M.P.). India
banshi_patidar@yahoo.com

ABSTRACT : *Intrusion detection has, over the last few years, assumed paramount importance within the broad realm of network security and it has significant impact on wireless ad hoc networks. These networks do not have an underlying infrastructure and the network topology constantly changes. The inherently vulnerable characteristics of wireless ad hoc networks make them susceptible to attacks, and it may be difficult to control before any prevention works out. Secondly, with so much advancement in hacking, if attackers use sophisticated technology, they will eventually succeed in infiltrating the system. This makes it important to constantly or periodically monitor what is taking place on a system and look for suspicious behavior. Intrusion detection systems (IDS)monitor audit data, look for intrusions to the system, and initiate a proper response. In this paper, we present a method for determining critical path that use the distributed security scheme to find out the malicious node. The nodes with the help of the critical links will find out the malicious node using distributed security scheme and inform all the nodes about malicious node. The simulation results describe the details of the critical path test.*

Keywords: Mobile ad hoc network, Security, Intrusion detection, Intrusion detection system, Distributed security scheme

Received : 11 August 2010, Revised 22 September 2010, Accepted 29 September 2010

1. Introduction

Wireless ad hoc networks have been in focus within the wireless research community. Essentially these are networks that do not have an underlying fixed infrastructure. Mobile hosts "join" on the fly and create a network on their own. With the network topology changing dynamically and the lack of a centralized network management functionality, these networks tend to be vulnerable to a number of attacks.

Wireless ad hoc networks find application in military operations so that planes, tanks, and moving personnel can communicate. Rescue missions and emergency situations also find use for such networks. Other examples include virtual classrooms and conferences wherein people can set up a network on the spot through their laptops, PDAs, and other mobile devices, assuming they share the same physical medium such as direct sequence spread spectrum (DSSS) or frequency hopped spread spectrum (FHSS) [2].

Mobile ad hoc networks (MANETs) present a number of unique problems for Intrusion Detection Systems (IDS). Network traffic can be monitored on a wired network segment, but ad hoc nodes can only monitor network traffic within their observable radio transmission range. A wired network under a single administrative domain allows for discovery, repair, response, and forensics of suspicious nodes. A MANET is most likely not under a single administrative domain, making it difficult to perform any kind of centralized management or control. In an ad hoc network, malicious nodes may enter and leave the immediate radio transmission range at random intervals, may collude with other malicious nodes to disrupt network activity and avoid detection, or behave maliciously only intermittently, further complicating their detection [1].

Intrusion prevention measures such as authentication and encryption are not guaranteed to work all the time, which brings out the need to complement them with efficient intrusion detection and response. If an intrusion is detected quickly enough, the intruder can be ejected before any damage is done or any data is compromised. An effective IDS can not only serve as a deterrent acting to prevent intrusions but also provide information about intrusions to strengthen intrusion prevention measures [3].

The remainder of the paper is organized as follows: Section 2 presents the background review and related work that are important for the understanding of the material to follow. Section 3 introduces the IDS which save the critical path by malicious activities, Section 4 Simulation Details; section 5 Simulation Results, Section 6 Conclusion, Section 7 Future Work, Section 8 References.

2. Background And Related Work

A number of IDS techniques have been proposed in the research literature. Moreover, a number of trust building and cluster-based voting schemes have been proposed to enable the sharing and vetting of messages, and data, generated and gathered by IDS systems. Zhang and Lee describe a distributed and collaborative anomaly detection-based IDS for ad hoc networks [4, 5]. Tseng et al. describe an approach that involves the use of finite state machines for specifying correct AODV routing behavior and distributed network monitors for detecting run-time violation of the specifications [6]. Pirzada and McDonald present a method for building confidence measures of route trustworthiness without a central trust authority. The authors also present a concise summary of previous work in the area of establishing trust in ad-hoc networks [7]. Theodorakopoulos and Baras present a method for establishing trust metrics and evaluating trust [8]. Michiardi and Molva assign a value to the "reputation" of a node and use this information to identify misbehaving nodes and cooperate only withnodes with trusted reputations [9]. Albers and Camp couple a trust-based mechanism with a mobile agent based intrusion detection system, but do not discuss the security implications or overhead needed to secure the network and individual nodes from the mobile agentsthemselves [10].Sun, Wu and Pooch introduce a geographic zone-based intrusion detection framework that uses location-aware zone gateway nodes to collect and aggregate alerts from intra-zone nodes. Gateway nodes in neighboring zones can then further collaborate to perform intrusion detection tasks in a wider area and to attempt to reduce false positive alarms [11]. Some recent papers have addressed different approaches to the IDS in wireless and ad-hoc networks. [12,13,14,15].

3. IDS For Critical Path

There are four sub sections in which our IDS will work.

3.1 Critical path.

3.2 Critical path detection.

3.3 Malicious behavior model.

3.4 Apply distributed security scheme to save critical path from attack.

3.1 Critical Path

In this section first, we describe the definition of a critical path is a path whose failure or malicious behavior disconnects or significantly degrades the performance of the network. Three steps are required to detect whether a testing node shares a critical link with its neighbor. The first step is to temporarily modify the testing node's routing table to allow only one communication link to be operational at a time, while blocking communication through all others. The enabled communication link will be between the testing node and a node other than the node under test. Each communication link will be tested sequentially in this manner to determine if an alternative path to the link under test exists. If an alternative path exists, then the link is not critical because its removal will not disconnect the network. In order to temporarily change the routing tables, we route all the outgoing network traffic through the link shared with a neighbor node other than the node under test, and execute the following commands:

3.2 Critical Path Detection

In this simulation module we use the trace file generated by ns-2, which is used as an input for C++ structure file, where we have created two linked lists. One link list stores incoming node numbers and other outgoing node numbers. In this file we used a count variable as global variable. These count variable stores the total number of pair's m-n, where m and n are some positive integer. Whenever first value pair comes then the count variable will increase and in the same manner we read all the incoming and outgoing node number and set the count for this path. After that we check that which incoming and outgoing node count value is greater in all pair's and set the path between these nodes as a critical path. After that we enter a worm propagation model in the network and check the status of the network and then we find out the intruder node.

3.3 Malicious Behavior Model

In AODV, a malicious node can override the restriction put by *RREQ_RATELIMIT* (limit of initiating / forwarding RREQs) y increasing it or disabling it. A node can do so because of its self-control over its parameters. The default value for the *RREQ_RATELIMIT* is 10 as proposed by RFC 3561. A compromised node may choose to set the value of parameter *REQ_RATELIMIT* to a very high number.

This allows it to flood the network with fake RREQs [12] and lead to a kind of DoS attack. In this type of DoS attack a non-malicious node cannot fairly serve other nodes due to the network-load imposed by the fake RREQs. This leads to the following problems:

- Wastage of bandwidth.
- Wastage of nodes' processing time (more overhead).
- Exhaustion of the network resources like memory (routing table entries).
- Exhaustion of the node's battery power.

3.4 Distributed Security Scheme

3.4.1 Overview

As mentioned earlier, the default value for *RREQ_RATELIMIT* is 10 RREQs/sec. This means each node is expected to observe some self-control on the number of RREQs it sends in one sec. A compromised node may choose to set the value of parameter *RREQ_RATELIMIT* to a very high number or even disable this limiting feature, thus allowing it to send large number of RREQ packets per second. The proposed scheme shifts the responsibility to monitor this parameter on the node's neighbor, thus ensuring the compliance of this restriction. This solves all of the problems (mentioned in section 2) caused due to flooding of RREQs from a compromised node. Thus instead of self-control, the control exercised by a node's neighbor results in preventing the flooding of RREQs.

3.4.2 RREQ_ACCEPT_LIMIT and RREQ_BLACKLIST_LIMIT

The proposal is based on the application of two parameters: *RREQ_ACCEPT_LIMIT* and *RREQ_BLACKLIST_LIMIT*. *REQ_ACCEPT_LIMIT* denotes the number of RREQs that can be accepted and processed per unit time by a node. The purpose of this parameter is to specify a value that ensures uniform usage of a node's resources by its neighbors. RREQs exceeding this limit are dropped, but their timestampsare recorded. This information will aid in monitoring the neighbor's activities. In the simulations carried out, the value of this parameter was kept as three (i.e. three RREQs can be accepted per unit time). This value can be made adaptive, depending upon node metrics such as it memory, processing power, battery, etc.

The *RREQ_BLACKLIST_LIMIT* parameter is used to specify a value that aids in determining whether a node is acting malicious or not. To do so, the number of RREQs originated/forwarded by a neighboring node per unit time is tracked. If this count exceeds the value of *RREQ_BLACKLIST_LIMIT*, one can safely assume that the corresponding neighboring node is trying to flood the network with possibly fake RREQs. On identifying a neighboring node as malicious, it will be blacklisted. This will prevent further flooding of the fake RREQs in the network. The blacklisted node is ignored for a period of time given by *BLACKLIST_TIMEOUT* after which it is unblocked. The proposed scheme has the ability to block a node till *BLACKLIST_TIMEOUT* period on an incremental basis. The *BLACKLIST_TIMEOUT* period is doubled each time the node repeats its malicious behavior.

In the simulations the value of *RREQ_BLACKLIST_LIMIT* is kept as 10 (i.e. more than 10 RREQs per unit time results in flooding activity). By blacklisting a malicious node, all eighbors of the malicious node restrict the RREQ flooding. Also the malicious node is isolated due to this distributed defense and so cannot hog its neighbor's resources. The neighboring nodes are therefore free to entertain the RREQs from other genuine nodes. Nodes that are confident about the malicious nature of a particular node, can avoid using it for subsequent network functions. In this way genuine nodes are saved from experiencing the DoS attack.

4. Simulation Details

The simulation described in this paper was tested using the ns-2 test-bed that allows users to create arbitrary network topologies [12]. By changing the logical topology of the network, ns-2 users can conduct tests in an ad hoc network without having to physically move the nodes. ns-2 controls the test scenarios through a wired interface, while the ad hoc nodes communicate through a wireless interface.

we first generate the test traffic using tcl script and find out the critical link in the network and then worm propagation model is injected in the network through critical link and IDS find out the attack (DoS attack route request type attack in our case) using distributed security scheme. And then find out the experimental results for TCP, UDP and CBR traffic in three cases.

1> UDP packets at receiving end compare with actual transmitted packet from sender end without malicious activities.

2> UDP packets at receiving end compare with actual transmitted packet from sender end during malicious activities.

In order to illustrate the detection of critical path,we first generate some test traffic in the network. TCP, UDP and CBR traffic is generated from some nodes and some nodes in network are received these traffic. Like nodes 0 and 1 are generating TCP traffic and node 3 and 4 are receiving this traffic.

The Simulation ends up with generation of a trace file and a nam file. We use the trace file to retrieve **Hs (**id for this source node) and **Hd (**id for next hop towards the destination). AWK utility is used to
retrieve those fields from trace file and save them in a different file then the generated output file can be use for calculating number of packet travel from each path by using C++, after that we find out the path from where the maximum data has travel and set the path as a critical path.

FROM _NODE	TO_NODE	NO OF PACKET RECORDED
6	**4**	**138**
9	**7**	**138**
4	**3**	**462**
5	**4**	**328**
7	**6**	**540**
FROM_NODE	**TO_NODE**	**MAXIMUM TRAFFIC**
7	**6**	**540**
TOTAL TRAFFIC FROMALL NODES		**3601**

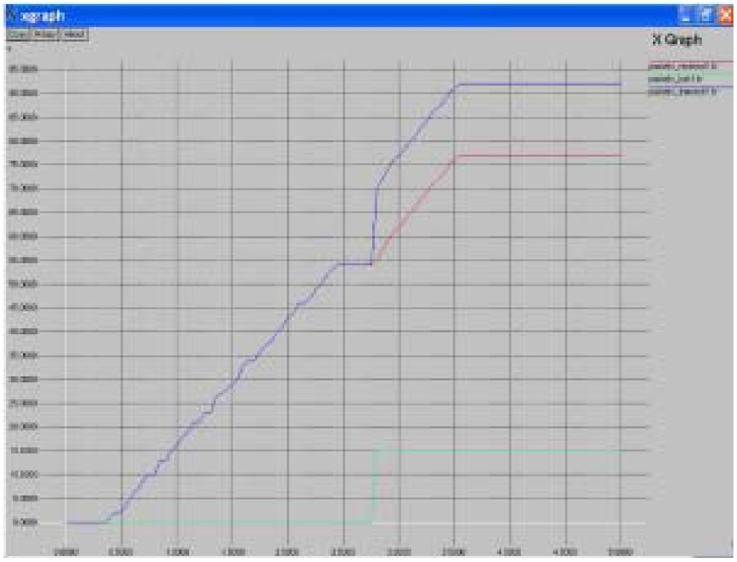
Table 1. shows the source to destination path and traffic.

Table 1 shows the source to destination path and traffic Then after detecting the critical path a malicious node comes into the radio range of critical path's end nodes (critical nodes they are the end nodes of critical link) and then generate the RREQ packet for unknown destination and continuous it will generate these packets to affect the critical path activities. The test traffic is again generated and then we again monitor the network and record the changes in the network. Changes can be inspected by examining the trace file. If we found any change in the information in any field of the trace file or more losses in CP, UDP and CBR raffic packet then we can conclude that the path is critical and our assumption of critical path is true otherwise it is not the critical path and we keep on checking the same thing for different paths. Once the malicious node comes into the radio range of critical nodes then both node will be affected by RREQ packet attack which will be continuously broadcasted by malicious node. Then apply distributed security scheme firstly used by critical nodes to detect the malicious node activities whenever they will find out the malicious behavior of node then they response the network about malicious node. After completion ofresponse phase all the node has identity about malicious node and they will not create any route to any destination in which this malicious node will exits. After applying IDS method again test traffic will generate and we find out that the performance will be increase in the presence of malicious node.

5. Experimental Results from Simulation

5.1 UDP Comparison before Intrusion

Figure 5.1 shows UDP packet transmission, packet receives and packet lost before intrusion. Packet lost is much lower than packet received

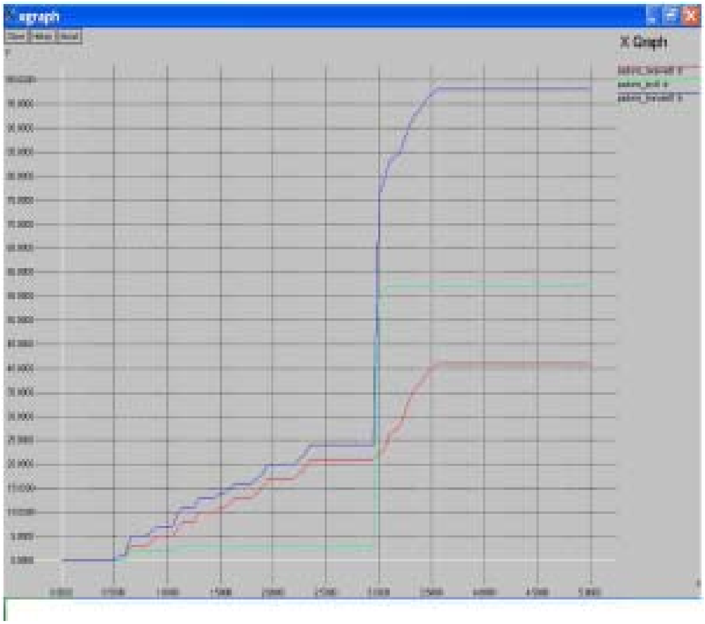


Time

Figure 5.1 UDP packets before intrusion

5.2 UDP Comparison After intruder

Figure 5.2 shows UDP packet transmission, packet receives and packet lost after intrusion. Packet lost is much lower than packet received.

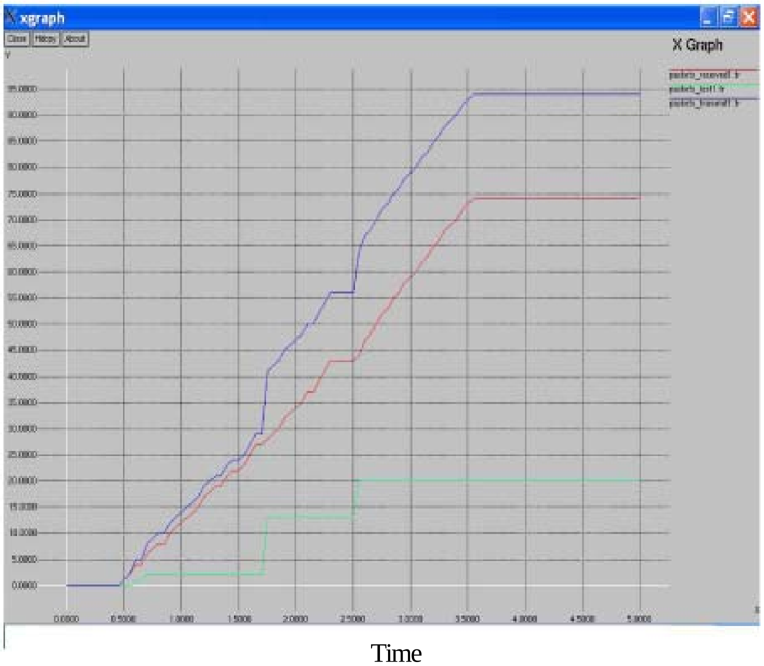


Time

Figure 5.2 UDP packets after intrusion

5.3 UDP Comparison After intrusion detection

Figure 5.3 shows UDP packet transmission, packet receives and packet lost after intrusion detection. Packet lost is much lower than packet received.



Time

Figure 5.3 UDP packets after intrusion detection

6. Conclusion

We perform number of test in ns-2 simulator and find out critical path after that we block the link and worn model infects the network. Here some result is shown.

UDP Packets					
Before intrusion		After intrusion		After intrusion detection	
Received%	Loss%	Received%	Loss%	Received%	Loss%
83.69	16.31	58.17	41.83	78.72	21.28

Table 2. Packet Comparison before and after intrusion

Based on number of simulation analysis, first we find critical path in ad-hoc network. And after that we check the activities of critical node by injecting the worm packets in that critical node and than analyze the UDP packets. And we find out the following information that shows when intruder comes in the network.

7. Future Work

In this project we assume less mobility of nodes and detect single critical path and inject this critical link by DoS RREQ ttack and apply distributed security scheme to identify the malicious activities of node and inform all the nodes of network about this malicious node. In future we trace more than one critical paths and nodes and also apply a new algorithm to work in frequently changed topological environment. We can not use this algorithm where mobility of node is high because in this algorithm there is less overhead and it will decrease the performance of network. We can also apply algorithm to find out various malicious activities like packet capturing, false route forwarding, changing source and destination addresses etc.

References

[1] Karygiannis, A., Antonakakis, E., Apostolopoulos, A. Detecting Critical Nodes for MANET Intrusion Detection Systems, Available at http://www.arygiannis@nist.gov

[2] Mishra, Amitabh., Nadkarni, Ketan., Patcha,Animesh (2004). Intrusion detection in wireless ad hoc network, IEEE wireless communication.

[3] Mishra, Amitabh.,Nadkarni, Ketan M. (2003). Security in Wireless Ad Hoc Networks, in Book *The Handbook of Ad Hoc Wireless Networks.* CRC Press LLC.

[4] Zhang, Y., Lee, W. (2000).Intrusion detection in wireless ad hoc networks, *In*: Proceedings of the 6th annual international conference on Mobile computing and networking, p. 275– 283. ACM Press.

[5] Zhang, Y., Lee, W., Huang, Y. (2002). Intrusion detection techniques for mobile wireless networks, ACM/Kluwer Mobile Networks and Applications (MONET).

[6] Tseng, C, Y., Balasubramanyam, P., Ko, C., Limprasittiporn, R., Rowe, Levitt, K. (2003). A specification based intrusion detection system for AODV, *In*: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, p. 125–134. ACM Press.

[7] Pirzada, Asad Amir, and McDonald, Chris. (2004) Establishing trust in pure ad-hoc networks, *In* Proceedings of the 27th conference on Australasian Computer Science - V 26, p 47-54.

[8] Theodorakopoulos, George., Baras, John.(2004) .Trust evaluation in ad-hoc networks, *In*: Proceedings of the 2004 ACM workshop on Wireless security, p. 1-10.

[9] Michiardi, P., Molva, R. (2002). Core: A Collaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks, Communication and Multimedia Security 2002 Conference.

[10]Albers, Patrick.,Camp, Olivier (2002). Security in Ad hoc Networks: a general Intrusion detection architecture enhancing trust based approaches. Proceedings of the First International Workshop on Wireless Information Systems2002.

[11] Sun, Bo, Wu, Kui and Pooch, Udo. (2003). Alert aggregation in mobile ad hoc networks". Proceedings of the 2003 ACM workshop on Wireless security, p.69 – 78.

[12] Umang, S. Reddy, B.V.R. Hoda, M.N. (2010). Enhanced intrusion detection system for malicious node detection in ad hoc routing protocols using minimal energy consumption, Communications, IET. 4 (17) 2084 – 2094.

[13] Mamun,Mohammad Saiful Islam., Sultanul Kabirm A.F.M. (2010). Hierarchical Design based Intrusion Detection System for Wireless Adho Sensor Network, *International Journal of Network Security & Its Applications (IJNSA)*, 12 (3) July 2010. 102-117.

[14] Khan, Shafiullah., Loo,Kok-Keong., Din,Zia Ud (2010). Framework for Intrusion Detection in IEEE 802.11 Wireless Mesh Networks, *The International* Arab Journal of Information Technology,7 (4) 435- 440.

[15] Mitrokotsa, Aikaterini., Mavropodi, Rosa., Douligeris, Christos (2006). Intrusion Detection of Packet Dropping Attacks inMobile Ad Hoc Networks, International Conference on Intelligent Systems And Computing: Theory And Applications. 111-118.

[16] Ping,, Yi ¨,Xinghao, Jiang., Yue,Wu., Ning, Liu (2008).Distributed intrusion detection for mobile ad hoc networks, *Journal of Systems Engineering and Electronics,* 19 (4) 851-859.

[17] The Network Simulator – ns-2 http://www.isi.edu/nsnam/ns/