Application Based Detection Technique for Secure Mobile Ad-hoc Network

Kamini Maheshwar, Sapna Bagde, Deshraj Ahirwar

Abstract-* An ad-hoc network is often defined as an infrastructure less network, meaning a network without the usual routing infrastructure like fixed routers and routing backbones. Typically, the ad-hoc nodes are mobile and the underlying communication medium is wireless. In mobile ad-hoc networks, the data tends to be intercepted by malicious node when using a single path for transmission. Also, the wireless channel in a mobile ad-hoc network is accessible to both legitimate network users and malicious attackers. So, the task of finding good solutions for these challenges plays a critical role in achieving the eventual success of mobile ad-hoc networks. In this paper, we proposed an efficient monitoring technique that uses readily available information from different layers of the protocol stack to detect “malicious packet-dropping”, where a faulty node silently drops packets destined for some other node. A key source of information for this technique is the messages used by the special ad-hoc routing protocols. This technique can be deployed on any single node in the network without relying on the cooperation of other nodes, easing its deployment. Our simulation results show that proposed technique has good detection effectiveness across a wide variety of network mobility models.

Keywords-* MANET, Secure Routing Protocol, Monitoring Detection Technique.

I. INTRODUCTION

Ad hoc networks are experiencing a major surge in interest in places where the fixed infrastructure is nonexistent, damaged, or impractical. In the absence of infrastructure, what is needed is that the wireless devices themselves take on the missing functions.

Mobile computers and applications will become indispensable in such situations. The wide deployment of the Internet has provided additional impetus for exploring the benefits of computer internetworking even in situations where neither the Internet nor any other internetwork is reachable. In such situations, one might wish to use familiar network programs to carry on the same kinds of interactive computing with neighbors and associates in the area. The third characteristic of an ad-hoc network implies that every node in an ad-hoc network volunteers to forward packets on behalf of other nodes. It is this node cooperation that holds an ad-hoc network together and makes the communication among the nodes possible.

But such node cooperation cannot always be taken for granted. There could be situations in which a node might refuse to cooperate. Some of the reasons might be genuine while others indicate malicious or selfish intent. Some possible reasons for a node’s non-cooperation include:

→*Low Battery:* Nodes with reduced battery power might limit their activities to periodically transmitting and receiving emergency or high-priority messages to conserve the remaining battery power and thus extend their duration of operation.

→ *Malicious Intent:* A node might want to disrupt the communication by misrouting, dropping or corrupting data packets. This scenario is very likely to occur in battlefield operations where the enemy nodes are always trying to disrupt the ongoing communication.

→*Selfish Behavior:* Every node in an ad-hoc network must forward packets on behalf of others even if they are not of interest to it. So, a node might not be willing to expend its battery power on behalf of others [1] and [3] and [4].

In any case, it is imperative to detect such behavior and take appropriate action to avoid any unnecessary wastage of scarce network resources like bandwidth, battery power, etc to retransmit the packets and to exchange control information. In other words, such behavior impedes the efficient functioning of the ad-hoc network. Detecting malicious behavior is the very first step in handling malicious nodes. Once malicious behavior is detected, the next step would be to identify the misbehaving node(s) in the ad-hoc network and then to finally isolate them so that the ad-hoc network can start functioning in accordance with its intended purpose without any performance hit. In this paper, we proposed an efficient monitoring technique which does not require modification to all the nodes in the network and relies on readily available information at different network levels to detect the presence of malicious nodes.

II. BACKGROUND

Wireless Networks-

Wireless networks are experiencing unprecedented growth in the recent years. The primary reason being the greater user convenience promised by mobile computing. The wide proliferation of laptops, PDAs, and mobile phones means that there are an increasing number of devices on the move and hence there is a greater need to support such devices. In the wired networking world, a static network infrastructure is implicitly assumed to exist, with a host having the same point of attachment into the larger network over time. The user convenience promised by wireless networks allows a node to change its point of attachment over time and requires a number of additions to the existing network-layer architecture.

Manuscript received December 09, 2011.

Kamini Maheshwar, Computer Science & Engineering, BUIT Barkatullah University, Bhopal(MP), India, Phone/ Mobile No.09893719300, (e-mail: kaminimaheshwar01@gmail.com).

Sapna Bagde, Information Technology, BUIT Barkatullah University, Bhopal(MP), India, Phone/ Mobile No.09617233992, (e-mail: sbagde26@gmail.com).

Deshraj Ahirwar, Computer Science & Engineering, Samrat Ashok Technological Institute , Vidisha(MP), India, Phone/ Mobile,NO.09993795773,,(email:deshrajahirwar.sati@gmail.com).

In order to allow a mobile device maintain a connection with a remote application as the device changes its point of attachment due to mobility, it is necessary for the device to maintain its IP address. *Mobile IP* provides this transparency, allowing a mobile node to maintain its permanent IP address while moving among networks. But on the other hand, if such a connection maintenance is not required across points of attachment, a device can be assigned different IP addresses at different networks. This kind of functionality is already provided by the Dynamic Host Configuration Protocol (DHCP) [2].

Mobility Management-

In wireless networking, a mobile node has a permanent "home" known as the *home network*. The entity within the home network that performs the mobility management functions is known as the *home agent*. The network in which the mobile node is currently residing in is known as *foreign network*, and the entity within the foreign network that helps the mobile node with mobility management functions is known as a *foreign agent*. A *correspondent* is the entity wishing to communicate with the mobile node.

When a mobile node is resident in a foreign network, all traffic addressed to the node's permanent address now needs to be routed to the foreign network. One way to handle this is for the foreign network to advertise to all other networks that the mobile node is resident in its network. But the problem with this approach is that of scalability. The routers may have to maintain forwarding table entries for potentially millions of mobile nodes. An alternative approach is to push mobility functionality to the network edge by having the home agent in the mobile node's home network track the foreign network in which the mobile node resides [1] and [3].

Ad Hoc Network Vulnerabilities-

Due to the dynamic nature of a mobile ad-hoc network, it suffers with frequent topology changes. The network topology may change rapidly and unpredictably and the connectivity among the terminals may vary with time. The mobile nodes in the network dynamically establish routing among themselves as they move about. Mobile ad-hoc networks therefore should be able to adapt the traffic and propagation conditions as well as the mobility patterns of the mobile network nodes. For most of the light-weight mobile terminals, the communication-related functions should be optimized to save unnecessary power consumption. Wireless ad-hoc networks pose a different challenge for designing power efficient systems. Due to the absence of an infrastructure, each node in an ad-hoc network also acts as a router. For an ad-hoc network to exist, nodes have to be at least in the reception mode most of the time. Ad hoc networks should be able to balance traffic load among nodes such that power constrained nodes can be put into a sleep mode while traffic is routed through other nodes. Another area of concern could be the selfishness of an individual node. Participation in an ad-hoc network requires a node to expend its battery power by forwarding packets on behalf of other nodes. A node might have to do this even if it doesn't originate any data destined for some other node(s) in the ad-hoc network. So, not all nodes might be willing to expend their resources for others [4] and [6].

Handling Malicious Nodes-

The presence of malicious nodes poses a grave threat to the very existence of an ad-hoc network. It is imperative to handle such nodes to prevent the legitimate nodes from being hit and to enable the ad-hoc network deliver its services. There are three main steps in handling a malicious node.

→*Detection:* The first step in handling a malicious node is to detect the presence of any malicious nodes. This is done by looking for any distinct or peculiar network behavior such as increased packet drops or TCP timeouts at the source node.

→*Identification:* Once the presence of malicious node(s) is detected, the next step is to identify the misbehaving nodes(s). For example, a trace route mechanism can be used to identify a malicious node. After the successful identification of misbehaving node(s), all the nodes participating in the ad-hoc network should be informed so that they can avoid those nodes in their communication routes.

→ *Isolation:* Once all the nodes in the ad-hoc network are aware of the malicious node(s), they can cooperate to isolate those nodes by denying to provide them with any kind of service (For example, denying packet forwarding on behalf of such nodes) [4] and [7].

Related Works-

Sergio Marti et al discuss two techniques, namely "watchdog" and "pathrater" to improve the throughput in MANETs in the presence of compromised nodes that agree to forward but fail to do so. Watchdog is used to detect and identify a malicious node, while the pathrater performs the job of isolating that node. Every node in the network includes both a watchdog and a pathrater. When a node forwards a packet, the node's watchdog verifies that the next node in the path also forwards the packet. The watchdog does this by listening promiscuously to the next node's transmissions, which requires the presence of bi-directional links. If the next node does not forward the packet, it is misbehaving. The watchdog detects misbehaving nodes. Every time a node fails to forward the packet, the watchdog increments the failure tallies. If the tally exceeds a certain threshold, it determines that the node is misbehaving; this node is then avoided using the pathrater. The pathrater combines knowledge of misbehaving nodes with link reliability data to pick the route most likely to be reliable. Each node maintains a rating for every other node it knows about in the network. It calculates a path metric by averaging the node ratings in the path [3] and [8].

Buchegger and Le Boudec work to overcome some of the problems associated with the watchdog and pathrater technique using a method called "Nodes Bearing Grudges". This protocol is composed of four components that are closely coupled together. Nodes start out trusting all other nodes in the network, but build grudges against nodes that exhibit malicious behavior. The monitor component monitors neighboring nodes to detect malicious behavior in a similar manner as the watchdog. If it detects any malicious behavior, it alerts the reputation system. The reputation system evaluates the alarm and determines if the event is significant. If the event is significant, the event count is incremented.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Once the count reaches some threshold, the reputation for the misbehaving node is reduced. When the reputation for the node gets low enough, the path manager is alerted. The path manager is similar to the pathrater. It adjusts the ranks of paths based on information about nodes in the path. If a path has a malicious node, that path is deleted to prevent routing through the malicious node. The path manager also ensures that the node does not forward data for malicious nodes. This prevents the problem of rewarding bad behavior. Finally, the trust manager handles interaction with other nodes through the use of special alarm messages. The trust manager has a trust table and a friends table. The trust table is used when processing incoming alarm messages and the friends table is used when sending alarm messages. If a malicious node is detected, the trust manager sends an alarm message to other nodes in the friends table so that they will avoid the malicious node. When the node receives an alarm message, it looks up the source node in the trust table to see how much it trusts the sender. The trust level controls how much weight the event in the alarm message is given. The event is weighted and passed on to the reputation system. Problems with bogus alert messages in the watchdog and pathrater system are avoided through the use of message authentication. This prevents malicious nodes from denouncing other nodes with forged alarms. Overall, the nodes bearing grudges method works well, but it is more difficult to implement than the watchdog and pathrater. Like watchdog and pathrater, it requires modification to all nodes in the network. Furthermore, it requires security associations between nodes to authenticate messages [2] and [7] and [9].

Park and Lee use route-based distributed packet filtering to prevent Denial of Service (DoS) attacks. This technique was developed for wired networks but may be adapted to ad-hoc wireless networks. Packet filtering works by placing filters at key points in the network, which perform routability checks on incoming packets. The routability checks determine if the packet is traversing a legitimate path between the source and destination addresses. In an ad-hoc wireless network, this can catch some malicious behavior, including misrouting of packets, impersonation attacks where the malicious node is not next to the impersonated node, and possibly some black hole routing protocols attacks. The routability checks require knowledge of valid routes in the network, which is difficult to determine due to the dynamic nature of the network. In some routing protocols, such as DSDV and CGSR, each node has a table with all valid routes in the network. With other source-routed ad-hoc routing protocols, such as DSR, the packet carries the full route between the source and destination, and this information can be used to check for valid routing. However some popular on-demand routing protocols, such as AODV, may not have this information at every node. For these routing protocols, an additional mechanism is necessary to build the table of valid routes, and this mechanism would be vulnerable to attack by malicious nodes. Even with the routing protocols where information on valid routes is available, each node may not have the most recent routing information. This may cause a node to think that a packet is not traveling on a valid route when it actually is. While distributed packet filtering only requires modification of some of the nodes in the network, it may still be difficult to modify enough of the nodes to make it useful. In an ad-hoc wireless network, picking key nodes to place packet filters isn’t easy. The traffic in the network may not be concentrated on a few key junctions so the packet filters may miss a lot of packets. Even if there are key junctions when the filters are placed, there is no guarantee that the nodes will remain at key junctions due to mobility [1] and [5] and [6].

Perfect ingress filtering is a variant of route-based distributed packet filtering that places filters on all nodes in the network. It may catch more packets sooner than route-based distributed packet filtering and is effective in preventing some types of attacks, such as DDoS. Since the filters are placed on all nodes, every packet will be examined. Perfect ingress filtering has similar drawbacks to route-based distributed packet filtering. Because the filters are placed on all nodes, it is even more difficult to deploy. It may not even be possible to deploy on some nodes that are very limited in processing capability.

Zhang and Lee described a distributed and cooperative intrusion detection model where every node in the network participates in intrusion detection and response. In this model, an Intrusion Detection System (IDS) agent runs at each mobile node, and performs local data collection and local detection, whereas cooperative detection and global intrusion response can be triggered when a node reports an anomaly. The authors consider two kinds of attack scenarios separately:

- Abnormal updates to routing tables
- Detecting abnormal activities in layers other than the routing layer

Each node does local intrusion detection independently, and neighboring nodes collaboratively work on a larger scale. Individual IDS agents placed on each and every node run independently and monitor local activities (including user, systems, and communication activities with in the radio range), detect intrusions from local traces, and initiate responses. Neighboring IDS agents cooperatively participate in global intrusion detection actions when an anomaly is detected in local data or if there is inconclusive evidence. The data collection module gathers local audit traces and activity logs that are used by the local detection engine to detect local anomaly. Detection methods that need broader data sets or require collaborations among local IDS agents use the cooperative detection engine. To prevent malicious node from forging alert messages, the messages are authenticated. In addition, it requires modification to all the nodes in the network [3] and [10].

In this paper, we proposed an efficient monitoring technique that uses readily available information from different layers of the protocol stack to detect “malicious packet-dropping”, where a faulty node silently drops packets destined for some other node. A key source of information for this technique is the messages used by the special ad-hoc routing protocols. This technique can be deployed on any single node in the network without relying on the cooperation of other nodes, easing its deployment.

III. PROPOSED TECHNIQUE

MANET is imperative to develop solutions that are scalable, implementable and capable of detecting malicious behavior while the communication is in progress. Now, an *efficient monitoring* technique is proposed to overcome some of the problems associated with the existing techniques.

This technique can be used to detect faults such as dropping or misrouting packets. The main focus of this research is *malicious packet-dropping*, where a node intentionally drops packets that are destined for other nodes. The methodology and the algorithm used for detecting malicious packet dropping is discussed. The unobtrusive monitoring technique relies on readily available information at different network levels to detect the presence of malicious nodes and does not require modification or cooperation of all the nodes in the network. This technique mainly involves collecting and analyzing locally available data. Local data such as DSR route request and route error messages, and TCP timeouts is used to detect malicious behavior in the network. Some of the salient features of this technique include:

→***Single node operation***: Unobtrusive monitoring requires modification only to the node that it runs on.

→ ***Portable***: This technique does not require any new protocols. It works with existing protocols, such as DSR, mobile IP, and ICMP which allows the technique to be easily ported to many different systems.

→***No additional battery wastage:*** This technique uses data that is readily available in the network. So, it does not dissipate or waste battery power for exchanging control information with the neighboring nodes.

→***No node cooperation***: This approach does not rely on the cooperation of other nodes in the network.

→***No security associations***: since this technique does not need the cooperation of other nodes in the network, there is a requirement to have security associations between the nodes.

Other security mechanisms such as "Nodes Bearing Grudges" and "Intrusion Detection in Wireless Ad-Hoc Networks" require security associations between neighboring nodes to authenticate the messages passed among them.

→ ***No infrastructure***: It does not require support of any type of infrastructure, such as network controllers or certificate authorities.

→***Highly scalable***: Since this technique is not tied down by the cooperation or security associations between neighboring nodes, it can be incorporated into as many nodes as needed making it highly scalable.

Currently, unobtrusive monitoring has been tested to work with DSR, and it is expected to work with other routing protocols as well. The unobtrusive monitoring technique uses data that is readily available from different network levels. The data collection and analyser components lie at the core of the detection technique. The data collection component collects useful control information such as DSR Route Error messages and TCP timeout and retransmission times. The data collection component gathers this information received within a certain interval of time called the *detection interval*. Any information older than the detection interval is discarded which guarantees the freshness and relevance of the collected information and also suits the requirements of a memory constrained node. This collected data is passed on to the data analyser component which extracts useful information from these control messages and checks for any deviation from normal behavior. The information extracted by the analyser may include the following:

→ The TCP flow on which the DSR route error message is received.

→ The TCP flow id on which a packet timed out and the sequence number of that packet.

→The time that each message was received or each event occurred.

The data analyser uses this information to determine if any malicious activity is taking place. If any such behavior is detected, the corresponding node is alerted so that it can take appropriate action.

Algorithm 1: Data Collection (*detection interval)*

```
//
For;; do
  if DSR Route Error Message Received then
    fid: = Flow on which the route error was received;
    Store the received route error in the store corresponding to fid
    current time:= get current time;
    if there are messages older than "current time – detection interval" then
      purge those messages from the store;
    end if
  end if
end if
//
```

The main function of the "data collection" component is to record all the route errors received on a per flow basis at the source node. The collection component waits till it receives any route error message. Then it extracts pertinent information from the packet and records the occurrence on a per flow basis. If there exist any route error messages older than that allowed because of the "detection interval", it purges all those messages so that the freshness and relevance of the information is maintained. The information gathered by the data collection component is given to the data analyzer component whenever it detects a TCP timeout at that source.

Algorithm 2: Data Analyser (*detection interval)*

```
//
for ;; do
  if TCP timeout occurred then
    fid:= Flow on which the timeout occurred;
    current time := get current time;
    if any route error messages received for fid then
      if no route error messages in [current time – detection interval, current time] then
        raise a flag indicating malicious activity;
      end if
    else
      raise a flag indicating malicious activity;
    end if
  end if
end for
//
```

The "data analyzer" component waits for the occurrence of a TCP timeout at the source node. Whenever a timeout occurs, it obtains the corresponding flow on which this timeout occurred and obtains the route error information for that flow from the data collection component. It then tries to correlate the timeout with any of the route error messages recorded by the data collection component.

If it fails to find a route error message within the given "detection interval", it raises a flag informing the source node of a possible malicious behavior in the corresponding flow. The source node can then use this information to take any corrective action for the detected malicious behavior.

The main assumptions being made in the unobtrusive monitoring technique include:

→ The ad-hoc network has DSR as the underlying routing protocol and TCP is the underlying transport layer protocol.

→A node chosen to behave maliciously does so starting at some random time and from that time onwards it drops all the packets it receives.

→A malicious node only drops packets that belong to the higher layers (for eg. tcp) in the TCP/IP stack but not the control messages sent out by the DSR agents for route discovery, maintenance etc.

→Malicious nodes do not collude with one another.

→All the packet drops are either due to malicious packet dropping or due to broken link(s) at some intermediate node in the source route.

→All the mobile nodes have enough memory to store information about the Route Error messages received within the detection interval.

Detection interval is the duration within which the data collection component gathers control information from different levels of the network stack (which is finally fed to the data analyzer component) and purges any old information that falls outside the interval. As explained earlier, the length of the detection interval plays a vital role in determining the *detection efficiency* and *false positive rate* of the unobtrusive monitoring technique. A lower detection interval might cause genuine route error messages to be purged, raise false alarms when the TCP timesout and thereby increasing the false positive rate of the technique. Similarly, a higher detection interval also might have a negative effect on the detection technique. A higher detection interval might result in failing to detect malicious behavior by associating any TCP timeouts with old route error messages. Therefore, a higher interval lowers the detection efficiency of the technique. The effect of detection interval on the detection effectiveness and false positive rate of our unobtrusive monitoring technique will be discussed in more detail in the "Simulation Results" chapter. We will also present the detection effectiveness and the false positive rate of our technique for different mobility models. Also, if a malicious node selectively drops the packets instead of dropping all of them, it will be hard for the source node to act upon the alarms raised by the data analyser component. For example, if the source node acts on the alarms only after the number exceeds some threshold value within a specific time interval, the malicious node could spread out its malicious drops so that the number of drops does not exceed the threshold within the time interval, making it difficult for the source node to act on them. It is also possible for malicious nodes to collude with one another to launch sophisticated attacks without being detected.

The primary metrics used for evaluating the performance of the proposed unobtrusive monitoring technique are detection effectiveness and false positive rate.

Detection Effectiveness: The algorithm used for calculating the "detection effectiveness" of our approach is as given below:

Algorithm 3: Detection Effectiveness (det intval)

```
//
for each TCP timeout TO do
    if TO due to malicious packet drop or corresponding acknowledgement drop then
        recorded ++;
        src:= Source node at which timeout occurred;
        fid= Flow on which the timed out packet was sent;
        retrans= Time at which the packet was retransmitted;
        route error found:= route error exists [retrans - detinval, retrans];
        if route error found == false then
            detected++;
        end if
    end if
end for
return detected/recorded; //
```

IV. SIMULATION RESULTS

The choice of the detection interval determines the detection effectiveness and false positive rate of the proposed monitoring technique. Increasing the detection interval might allow unrelated Route error messages to be associated with any TCP time out at the source node and hence hampering the ability of the technique to detect malicious activity. Having a very small detection interval also has a negative effect on the false positive rate of the technique. If we have a smaller detection interval, we might quickly jump to a conclusion about a TCP time out at the source without waiting long enough to include any delayed Route Error message(s) for that flow and hence increasing the false positive rate of the technique. Also, increasing the detection interval means that a node has to store information for a longer period of time. If the node is receiving a lot of messages, this can drastically increase the storage overhead which is a burden on the memory constrained mobile nodes. Therefore, choice of the detection interval has a very significant impact on the performance metrics and storage overhead of the technique. In all our simulations, we have experimented with detection intervals ranging from 5 to 50 seconds in 5 second increments. We observed that at a lower detection interval, we have high detection effectiveness and also a high false positive rate. For the detection intervals of 30, 35, and 40 seconds our technique had good detection effectiveness with a lower false positive rate. To show how the detection effectiveness and false positive rate vary with the choice of detection interval, we present the detection effectiveness and false positive rate for high mobility random way point mobility networks for the following detection intervals: 20, 30, 35, 40, and 50 seconds. For the remaining models, we present the results only for detection intervals of 30, 35, and 40 seconds. The different configuration parameters used for generating the required mobility scenarios as Random Way Point (Medium Mobility), Random Way Point (High Mobility), RPGM (Medium Mobility), RPGM (High Mobility).

Random Way Point Mobility Networks-

In these kinds of networks a node chooses a random destination, speed and starts moving towards that destination. Between movements it pauses for some amount of time referred to as "pause time".

To simulate medium mobility networks a maximum node speed of 5 meters/second and a pause time of 30 seconds were chosen and a maximum speed of 20 meters/second and a pause time of 5 seconds were chosen for high mobility networks. For high mobility networks, we present the results for detection intervals of 20, 30, 35, 40, and 50 seconds. For medium mobility networks, we present the results for detection intervals of 30, 35, and 40 seconds. Also, for the metrics considered, the number of malicious nodes is increases from 5% to 40% in 5% increments.

→***Detection Efficiency*:**

The detection effectiveness for medium and high mobility networks is as shown in figures 1 and 2 respectively. From these figures we can see that for both medium and high mobility networks, increasing the detection interval lowers the detection effectiveness. This is because when using a bigger detection interval, there is a much higher probability of getting unrelated route error messages within this interval.

Also, we can see that the detection effectiveness is better in the case of medium mobility networks when compared to high mobility ones. Due to higher mobility, the links get broken quite frequently and there are many route error messages sent out by the nodes in the network. This also increases the probability of receiving unrelated route error messages within the detection interval at a source node and the source node correlates any TCP timeouts with the received route error message and thus leads to a decrease in detection effectiveness.

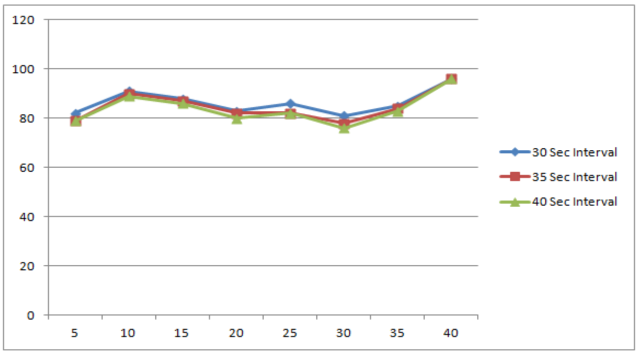
And also analyze the false positive rate for medium and high mobility networks. We can see that increasing the detection interval lowers the number of false positives. This is because, when we have a bigger interval we wait long enough to receive any delayed route error messages and do not quickly jump to conclusions. As the nodes move faster, the links between the nodes get broken all the more. So, the route error messages sent out by intermediate nodes might get dropped before they reach the source node. When the source node timesout it will not find any related route error messages received by it within the detection interval and might come to a conclusion that the timeout is due to some malicious behavior in the source route. So, at higher mobility there could be a higher false positive rate due to the loss of route error messages.

Reference Point Group Mobility

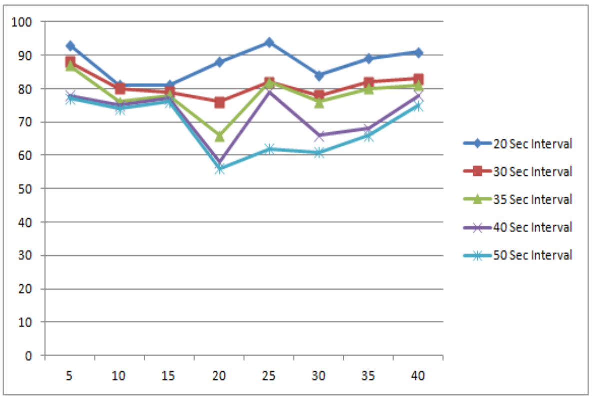
→ ***Detection Efficiency*:**

The detection effectiveness of the unobtrusive monitoring technique for “Reference Point Group Mobility” is as shown in figures 3 and 4 for medium and high mobility scenarios respectively.

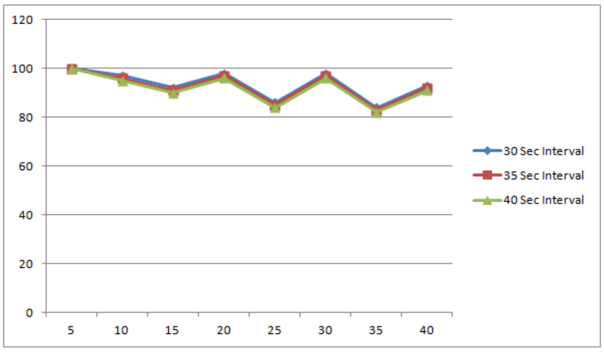
The analysis provided for the detection effectiveness of “Random Way Point” networks holds true here also. We can observe that increasing the detection interval decreases the detection effectiveness as in the case of random way point mobility networks. This decrease could be attributed to the unrelated route error messages being considered for correlation with the TCP timeouts when using a higher interval.



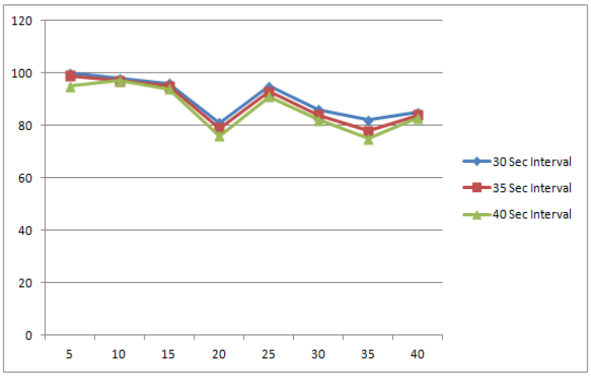
Detection Effectiveness vs. Percent of malicious Nodes
Figure1: Detection Efficiency – Random Way Point (Medium Mobility)



Detection Effectiveness vs. Percent of malicious Nodes
Figure 2: Detection Efficiency – Random Way Point (High Mobility)



Detection Effectiveness vs. Percent of malicious Nodes
Figure 3: Detection Efficiency – Reference Point Group Mobility (Medium Mobility)



Detection Effectiveness vs. Percent of malicious Nodes
Figure 4: Detection Efficiency – Reference Point Group Mobility (High Mobility)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

We can also observe a trend similar to random way point networks when we go from medium to high mobility. So, with the increase in mobility there will be more broken links and more route error messages sent out to the source nodes. So, the source node could miss some malicious drop events by correlating a TCP timeout with one of the many route error messages received by it due to increased mobility in the network. Also, since the nodes move in groups and with reference to a leader, there will not be as many broken links as in the case of random way point mobility. So, the chance of getting a stray route error message within the detection interval is much lower in the case of reference point group mobility when compared to random way point. So, in this case we can expect a much higher detection efficiency when compared to the random way point mobility model. The false positive rate of the unobtrusive monitoring technique for "Reference Point Group Mobility" is also, the increase in mobility affects the false positive rate of the technique. It is more likely for the route error messages to be dropped by some intermediate nodes before they actually reach the intended source node. So, the source node, which is unaware of the loss of route error messages, attributes any genuine timeouts due to broken links as being malicious. This leads to an increase in the false positive rate at higher mobility. Also, in the case of reference point group mobility, there will be fewer broken links when compared to the random wap point mobility due to the nature of movement of the nodes. So, misdetection a single normal timeout as malicious will greatly affect the false positive rate.

V. CONCLUSION AND FUTURE WORKS

Mobile ad-hoc networks constitute an emerging wireless networking technology for future mobile communications. However, unless the networks can be secured against malicious activity, their usefulness may be stifled. The task of finding good solutions for these security challenges prevalent in ad-hoc wireless networks will play a critical role in achieving the eventual success and potential of mobile ad-hoc network technology. To help protect ad-hoc wireless networks from malicious nodes, we developed an unobtrusive monitoring technique to detect malicious behavior in the network by gathering information from different network levels without relying on node cooperation. Unlike some other proposed methods, this technique is easy to deploy, since it only requires modification to a single device, and it does not require any additional infrastructure or security associations. Simulation results show that this technique has good detection effectiveness across a wide variety of network mobility models. The detection effectiveness tends to decrease when the network is highly loaded, when there is a long distance between neighboring nodes, or when the nodes are highly mobile. These situations are problematic for the network in general, since they cause increase in route maintenance and a decrease in packet transmission success. This technique also maintains low false positive rate in all the different scenarios considered. In the future, we would like to extend the unobtrusive monitoring technique to distinguish packet drops arising due to congestion and malicious behavior. We also plan to investigate the use of this technique with other ad-hoc routing protocols, such as AODV, TORA, and with other types of networks, such as hybrid wired-wireless networks and traditional wired networks.

REFERENCES

1. S.Tamilarasan and Dr.Aramudan, "A Performance and Analysis of Misbehaving node in MANET using Intrusion Detection System", IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.5, May 2011, pp 258-264.
2. Vijay Kumar, "BEHAVIORAL STUDY OF DYNAMIC ROUTING PROTOCOLS FOR MANET", International Journal of Computing and Business Research ,Volume 2 Issue 2 May 2011.
3. G. Rajkumar and K. Duraisamy, "A Fault Tolerant Multipath Routing Protocol to Reduce Route Failures in Mobile Adhoc Networks", European Journal of Scientific Research, Vol.50 No.3 (2011), pp.394-404
4. Wenchao Huang, Yan Xiong, Depin Chen, "DAAODV: A Secure Ad-hoc Routing Protocol based on Direct Anonymous Attestation", IEEE 2009 International Conference on Computational Science and Engineering, Issue Date : 29-31 Aug. 2009, Volume : 2 , On page(s): 809.
5. A.H Azni, Azreen Azman, Madihah Mohd Saudi, AH Fauzi, DNF Awang Iskandar, "Analysis of Packets Abnormalities in Wireless Sensor Network" , IEEE 2009 Fifth International Conference on MEMS NANO, and Smart Systems, pp 259-264.
6. Cuirong Wang, Shuxin Cai, "AODVsec: A Multipath Routing Protocol in Ad-Hoc Networks for Improving Security", IEEE 2009 International Conference on Multimedia Information Networking and Security, pp 401-404.
7. A Nagaraju and B.Eswar, "Performance of Dominating Sets in AODV Routing protocol for MANETs", IEEE 2009 First International Conference on Networks & Communications, pp 166-170.
8. Sheng Cao and Yong Chen, "AN Intelligent MANet Routing Method MEC", 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp 831-834.
9. [9] Zeyad M. Alfawaer and Saleem Al_zoubi, "A proposed Security subsystem for Ad Hoc Wireless Networks", IEEE 2009 International Forum on Computer Science-Technology and Applications, pp 253-256.
10. Shayesteh Tabatabaei, "Multiple Criteria Routing Algorithms to Increase Durability Path in Mobile Ad hoc Networks", IEEE 2009 by the Institute of Electrical and Electronics Engineers, Issue Date : 9-12 Nov. 2009, On page(s): 1 , Print ISBN: 978-1-4244-5647-5, INSPEC Accession Number: 11135758.

AUTHOR'S PROFILE

Kamini Maheshwar was born in 1987. She is a PG Scholar in BUIT Barkatullah University Bhopal (MP). She has done his BE (CSE) from BUIT BU Bhopal (MP) . His Research areas include Network Security , Adhoc Network ,Routing protocol, Data Mining .

Sapna Bagde was born in 1986.She is a PG Scholar in BUIT Barkatullah University Bhopal (MP). She has done his BE (CSE) from BUIT BU Bhopal (MP) . His Research areas include Network Security , Adhoc Network ,Routing protocol, Data Mining .

Deshraj Ahirwar was born in 1987.He is a PG Scholar from SATI Vidisha (MP). He has done his BE (CSE) from BUIT BU Bhopal (MP) . His Research areas include Network Security, Adhoc Network ,Routing protocol, Data Mining .He has published 3 international papers and 3 national papers .