A Simulation Study of Wormhole Attack in NS3

Mohseen Mukaddam[1] Sanket Dighe[2] Akash Varude[3] Aditya Supugude[4] Vikas Sangle[5]
[1,2,3,4,5]B.Tech Student
[1,2,3,4,5]Department of Information Technology
[1,2,3,4,5]College of Engineering Pune, India

Abstract— The ubiquity and ease of operation have made wireless networks extremely popular. The Downfall of such popularity is the increased attacks on Mobile Ad-hoc Networks (MANET). The attack studied in this paper is wormhole attack. The choice of attack was based on the difficulty in identifying and mitigating the attack in real life scenarios. We used AODV protocol for the simulation; the most popular routing protocol used for small-scale networks. Wormhole attack is a potential threat to MANETs; it involves using a two-node system to route packets through a 'tunnel' created by the nodes. The attack enables the attacker to manipulate packet traffic, while making it difficult for an Intrusion Detection System (IDS) to detect.

Key words: AODV, MANET, NS3, Wormhole Attack

I. INTRODUCTION

MANETS (Mobile Ad-hoc Networks) is a collection of wireless mobile hosts without fixed network infrastructure and centralized administration. The nodes can communicate with each other if in range or can route the packets via intermediary nodes. These wireless nodes have an interface to communicate with each other. These networks are dynamic and resilient; can sustain and function in the absence of base station.

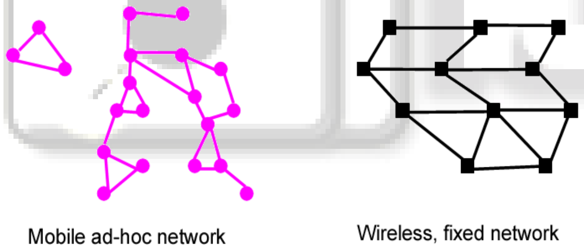


Fig. 1: MANETS vs. Fixed Networks

A. *MANET Characteristics:*

1) *Distributed operation:*

Unlike a fixed network, nodes distribute the control of the network. Frequent communications between nodes enable implementing special function like security and routing.

2) *Multi hop routing:*

Intermediate nodes relay the packets sent from nodes that are out of range.

3) *Autonomous terminal:*

Nodes do not depend on each other and perform the function of router and host.

4) *Dynamic topology:*

The freedom of movement for nodes means topology is dynamic; the nodes establish routing among themselves, creating their own network.

As mentioned earlier, each node acts a router that forwards data packets to other nodes. The routing protocol plays a huge role in determining network efficacy and packet delivery. Section II will describe types of routing protocols. Section III will elaborate on wormhole attacks. Section IV will shed light on how the simulation study of wormhole attack was carried out in NS3.

II. ROUTING PROTOCOLS

There are primary types of routing protocols are: Proactive protocols, Reactive protocols and Hybrid protocols [1], [2], [3]. Proactive protocols constantly updated, lists of destinations and routes are updated by periodical beacon messages. Reactive protocols respond on demand. Hybrid protocols combine the features of Reactive and Proactive protocols. Routing protocol determines the network delay, throughput, energy efficiency and other such performance parameters.

The following subsections will talk about working of Destination Sequenced Distance Vector (DSDV), Dynamic Source Routing (DSR) and Ad-hoc On Demand Distance Vector (AODV)

A. *Dynamic Source Routing*

DSR is a reactive protocol. Due to source routing, nodes do not rely on the routing table of intermediary nodes. Routes are cached and learned routes are used for routing packets. Its advantages lie in the fact that it is simple and does not frequently send beacon messages to nodes (at the cost of higher overhead). Missing or broken links are notified to source node. A large network with high mobility renders this protocol inefficient, as the overhead is dependent on the path.

B. *Destination-Sequenced Distance Vector*

DSDV is an early implementation to convert a wired routing protocol function on an ad-hoc wireless networks. This was achieved by incrementing the sequence number label of the routes. DSDV implements Bellman-Ford algorithm to solve the routing loop problem. Each node must contain information to other nodes in the network. New sequence number and frequent updates are needed for the packet transmission. Due to very frequent updates, it is not suitable for large mobile networks.

C. *Ad-Hoc On Demand Distance Vector*

AODV is a modified working model of DSDV without its very frequent updates. The nodes send route messages only when packet transmission is about to commence thus utilizing lower overhead without flooding the network. Although setup time and initial communication time is longer for AODV, it is much more efficient than DSDV for larger networks. It also successfully utilizes the concept of sequence number to determine newer paths to the destination.

III. WORKING OF AODV

The source node sends a RREQ packet to the nodes in the network before sending a packet. A typical RREP packet contains source identifier, destination identifier, source sequence number, broadcast id, destination sequence number and time to live fields. Nodes update their path information only when the received destination sequence number is greater than or equal to last value stored by the node along with smaller hop count. On receiving a RREQ, intermediate nodes either forward it or reply back with a valid route to the destination with RREP. As mentioned earlier, the validity of the route is checked with destination sequence number. The nodes with valid routes to destination or destination node are allowed to reply back (RREP) to the source node. If the node forwards the RREQ then it appends the previous node address along with its broadcast id. A timer is set to delete this entry in the event, no RREP are received before the time expires. On receiving a RREP, information about previous node is also stored. Hop Count represents the distance in hops from the source to destination.

The destination will be updated in the route table entry with new sequence number if [8]:

- Destination Sequence Number received from RREQ is greater than the existing value in the route table entry.
- The Sequence numbers are equal, but the incremented hop count is smaller than existing hop count.
- The Sequence number is unknown.

After the sequence value is updated, the node begins searching for a reverse route to source address. A route could be created or updated using source sequence number. A reverse route is updated or created if [8]:

- If Source Sequence Number received from RREQ is greater than the existing value in the route table entry, it is updated.
- The valid sequence number field is made true.
- The next hop in the routing table becomes the node from which RREQ was received.
- The value of hop count is copied from RREQ packet.

An RREP packet once received by a node within its lifetime considers the route to be valid. On receiving a RREP packet, it finds a reverse route and increments the hop count by one. Destination sequence number in the route table is updated if [8]:

- The Destination Sequence Number in the RREP is greater than existing value and the value is valid.
- The Sequence Numbers are same, but the incremented hop count is smaller than that of existing value.
- The sequence number is marked as invalid in the routing table.
- The sequence numbers are same, but the route is marked as inactive.

When the RREP reaches to the source node, it can now send the data packets through the route that is set up.

A node generates router error packet RRER in the following situations [8]:

- While transmitting the data, if it notices a link break for the next hop (neighbor) of an active route in its routing table. Here the node first makes the list of unreachable destinations along with unreachable neighbors in the routing table.
- If it receives a data packet that is to be sent to a destination node for which it does not have an active route.
- If it gets an RERR from a neighbor for one or more active routes.

RERR packets come into play when nodes move around in the network. RERR denotes unreachable nodes. Each node possesses a precursor list of neighbors that might use the node as next hop to a destination. This list information is acquired during RREP packet generation.

Source Address	Source Sequence	Broadcast ID	Destination Address	Destination Sequence	Hop Count

Table 1: RREQ field

Source Address	Destination Address	Destination Sequence	Hop Count	Lifetime

Table 2: RREP Field

IV. Wormhole Attack

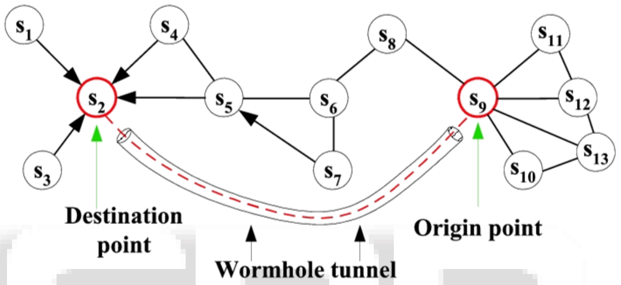


Fig. 2: Wormhole attack explained

Wormhole attack essentially creates a tunnel between two malicious nodes, thereby disrupting normal packet traffic. Figure 2 depicts a multi-node system, in which node S9 acts as the first end of wormhole tunnel and S2 acts as the second end of wormhole tunnel [4][5][7]. This tunnel is a direct line of communication between the two. One end will record all the packet information at one end and send it over to the other end. Needless to say, this compromises the security of the network; it also is difficult to detect being a two-node system [6]. By programming the nodes to behave maliciously for a certain time period and normally during the rest, increases the complexity of detecting such an attack. If the tunnel is created reliably, it can actually benefit the network, however that is not the intent of most malicious nodes.

V. Simulation

The Simulation was carried out in NS3 simulator version 3.21 with the following profile:

Parameters	Description
Examined Protocol	AODV
Simulation Time	1000 seconds
Simulation Area	150x150 m
Number of Nodes	13
Malicious Nodes	2
Network Traffic	CBR
Packet Size	512 Byte

Table 3: Simulation Profile

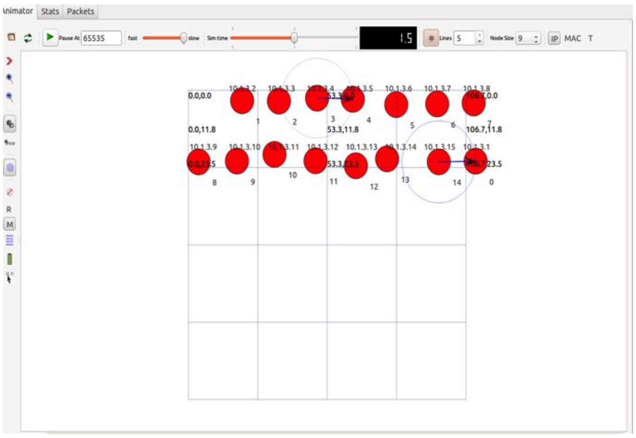
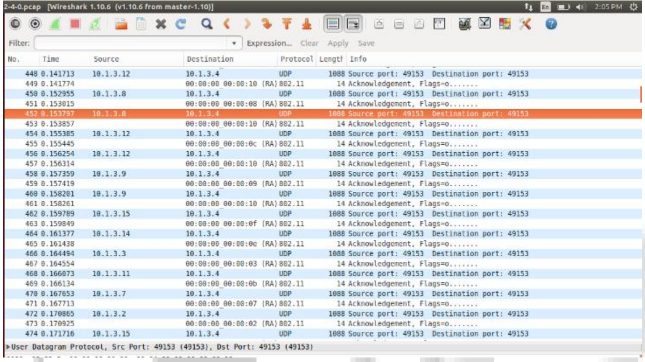


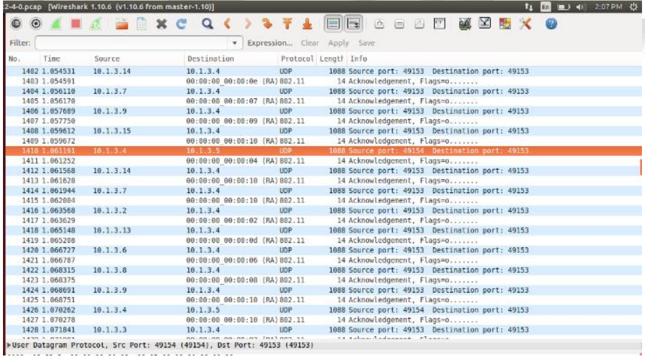
Fig. 3: Wormhole

In figure 3, a wormhole is created between node 3 (10.1.3.4) and node 4 (10.1.3.5). Under normal scenario, all the other nodes will transmit UDP packets to node 3 and none will interact with node 4. After some setup time, the wormhole is set up and node 3 will begin to forward these packets to node 4. The packet tracer demonstrates this:



```
2-4-0.pcap  [Wireshark 1.10.6  (v1.10.6 from master-1.10)]                                2:05 PM
Filter:                                  Expression...  Clear  Apply  Save
No.   Time      Source       Destination             Protocol Length Info
  448 0.141713  10.1.3.12    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  449 0.141774               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
  450 0.152955  10.1.3.8     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  451 0.153015               00:00:00_00:00:08 (RA)  802.11     14 Acknowledgement, Flags=o.......
  452 0.153797  10.1.3.8     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  453 0.153857               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
  454 0.155385  10.1.3.12    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  455 0.155445               00:00:00_00:00:0c (RA)  802.11     14 Acknowledgement, Flags=o.......
  456 0.156254  10.1.3.12    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  457 0.156314               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
  458 0.157359  10.1.3.9     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  459 0.157419               00:00:00_00:00:09 (RA)  802.11     14 Acknowledgement, Flags=o.......
  460 0.158201  10.1.3.9     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  461 0.158261               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
  462 0.159789  10.1.3.15    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  463 0.159849               00:00:00_00:00:0f (RA)  802.11     14 Acknowledgement, Flags=o.......
  464 0.161377  10.1.3.14    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  465 0.161438               00:00:00_00:00:0e (RA)  802.11     14 Acknowledgement, Flags=o.......
  466 0.164494  10.1.3.3     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  467 0.164554               00:00:00_00:00:03 (RA)  802.11     14 Acknowledgement, Flags=o.......
  468 0.166073  10.1.3.11    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  469 0.166134               00:00:00_00:00:0b (RA)  802.11     14 Acknowledgement, Flags=o.......
  470 0.167653  10.1.3.7     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  471 0.167713               00:00:00_00:00:07 (RA)  802.11     14 Acknowledgement, Flags=o.......
  472 0.170865  10.1.3.2     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
  473 0.170925               00:00:00_00:00:02 (RA)  802.11     14 Acknowledgement, Flags=o.......
  474 0.171716  10.1.3.15    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
▸User Datagram Protocol, Src Port: 49153 (49153), Dst Port: 49153 (49153)
```

Fig. 4: Normal UDP packet transmission



```
2-4-0.pcap  [Wireshark 1.10.6  (v1.10.6 from master-1.10)]                                2:07 PM
Filter:                                  Expression...  Clear  Apply  Save
No.   Time      Source       Destination             Protocol Length Info
 1402 1.054531  10.1.3.14    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1403 1.054591               00:00:00_00:00:0e (RA)  802.11     14 Acknowledgement, Flags=o.......
 1404 1.056110  10.1.3.7     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1405 1.056170               00:00:00_00:00:07 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1406 1.057689  10.1.3.9     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1407 1.057750               00:00:00_00:00:09 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1408 1.059612  10.1.3.15    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1409 1.059672               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1410 1.061191  10.1.3.4     10.1.3.5                UDP      1088 Source port: 49154  Destination port: 49153
 1411 1.061252               00:00:00_00:00:04 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1412 1.061568  10.1.3.14    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1413 1.061628               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1414 1.061944  10.1.3.7     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1415 1.062004               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1416 1.063568  10.1.3.2     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1417 1.063629               00:00:00_00:00:02 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1418 1.065148  10.1.3.13    10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1419 1.065208               00:00:00_00:00:0d (RA)  802.11     14 Acknowledgement, Flags=o.......
 1420 1.066727  10.1.3.6     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1421 1.066787               00:00:00_00:00:06 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1422 1.068315  10.1.3.8     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1423 1.068375               00:00:00_00:00:08 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1424 1.068691  10.1.3.9     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
 1425 1.068751               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1426 1.070262  10.1.3.4     10.1.3.5                UDP      1088 Source port: 49154  Destination port: 49153
 1427 1.070278               00:00:00_00:00:10 (RA)  802.11     14 Acknowledgement, Flags=o.......
 1428 1.071841  10.1.3.3     10.1.3.4                UDP      1088 Source port: 49153  Destination port: 49153
▸User Datagram Protocol, Src Port: 49154 (49154), Dst Port: 49153 (49153)
```

Fig. 5: Wormhole attack

The Figure 4 depicts normal traffic to node 3 (10.1.3.4), Figure 5 depicts Wormhole attack after some time for the same node.

The attack can lead to significant packet loss among other aftereffects of the attack. To carry out such a simulation, we had to override the packet forwarding logic of AODV protocol of NS3 for the wormhole ends.

VI. IMPLEMENTATION

As mentioned earlier, for the malicious nodes, the very behavior of the routing protocol had to be modified. An override function in the routing protocol was setup such that, if packet was forwarded to one of the end of the tunnel, then the destination would be overridden to the other end of the tunnel.

```
  void
  RoutingProtocol::RecvAodv (Ptr<Socket> socket)
  {
    NS_LOG_FUNCTION (this << socket);
    Address sourceAddress;
    Ptr<Packet> packet = socket->RecvFrom (sourceAddress);
    InetSocketAddress inetSourceAddr = InetSocketAddress::ConvertFrom (sourceAddress);
    Ipv4Address sender = inetSourceAddr.GetIpv4 ();
    Ipv4Address receiver = m_socketAddresses[socket].GetLocal ();
    NS_LOG_DEBUG ("AODV node " << this << " received a AODV packet from " << sender << " to " << receiver);
  //Override logic
  if(WrmAttack) {
        if(sender==FirstEndWormTunnel) {
        receiver=SecondEndWormTunnel;
}
        if(sender==SecondEndWormTunnel){
        receiver=FirstEndWormTunnel;
}
```

VII. CONCLUSION

The routing protocol used by MANETs must be reliable, secure, efficient and scalable. Security is a growing concern over the years and these routing protocols are no exception. AODV while being extremely popular is also vulnerable to attacks that involve modified Destination sequence number and hop count like the wormhole attack. The purpose of the study was to better understand the attack to help prevent it. There is undoubtedly more scope for research in this area, for securing routing protocols to theses flaws as well as intelligent Intrusion Detection Systems (IDS) that can weed out such attacks from the network.

REFERENCES

[1] Komala CR, Srinivas Shetty, Padmashree S., Elevarasi E., “Wireless Ad hoc Mobile Networks”, National Conference on Computing Communication and Technology, pp. 168- 174, 2010

[2] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu and Jorjeta Jetcheva, “A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols”, http://www.monarch.cs.cmu.edu/

[3] Perkins C. and Royer E. Ad hoc on-demand distance vector routing, In Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100 (1999)

[4] Harris Simaremare and Riri Fitri Sari. Performance Evaluation of AODV variants on DDOS, Blackhole and Malicious Attacks, International Journal of Computer Science and Network Security, VOL-11, June 2011, pp.6.

[5] K. Lakshmi, S.Manju Priya, A.Jeevarathinam, K.Rama and K. Thilagam. Modified AODV Protocol against Black hole Attacks in MANET, International Journal of Engineering and Technology Vol.2 (6), 2010.

[6] S Upadhyay . and B.K Chaurasia. Impact of Wormhole Attacks on MANETs, International Journal of

Computer Science & Emerging Technologies, Vol. 2, Issue 1, pp. 77-82 (2011)

[7] R. Maulik and N. Chaki. A Comprehensive Review on Wormhole Attacks in MANET. In Proceedings of 9th International Conference on Computer Information Systems and Industrial Management Applications, pp. 233-238, 2010

[8] MANET Routing Protocols and Wormhole Attack against AODV, International Journal of Computer Science and Network Security, Vol.10, No.4, April 2010.