Sinkhole Attack in AODV

Software: NetSim Standard v14.3, Visual Studio 2022

Project Download Link:

https://github.com/NetSim-TETCOS/Sinkhole-Attack-in-AODV-v14.3/archive/refs/heads/main.zip

Follow the instructions specified in the following link to download and setup the project in NetSim:

https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects

1 Introduction

Sinkhole attack is one of the most severe attacks in wireless ad hoc networks. In a sinkhole attack, a compromised or malicious node advertises false routing information to pretend itself as a specific node and receives whole network traffic. After receiving the whole network traffic, it can either modify the packet information or drop them to make the network complicated. Sinkhole attacks affect the performance of ad hoc network protocols such as the AODV protocol.
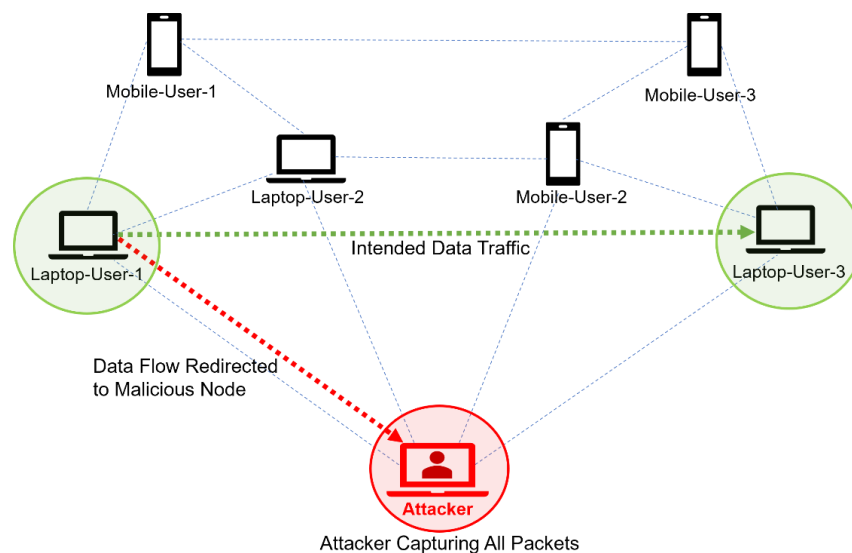

Figure 1 : Sinkhole attack in MANET using AODV

2 Implementation in AODV

- In AODV, the source broadcasts the RREQ packet during route discovery.

- The destination, upon receiving the RREQ packet, replies with an RREP packet containing the route to reach the destination.
- Intermediate nodes can also send RREP packets to the source if they have a route to the destination in their route cache.
- Taking advantage of this, a malicious node adds a fake route entry into its route cache, with the destination node as its next hop.
- Upon receiving the RREQ packet from the source, the malicious node sends a fake RREP packet with the fake route.
- The source node, upon receiving this fake RREP packet, perceives it as a better route to the destination.
- All network traffic is then attracted toward the sinkhole (malicious node), which can either modify the packet information or simply drop the packets.

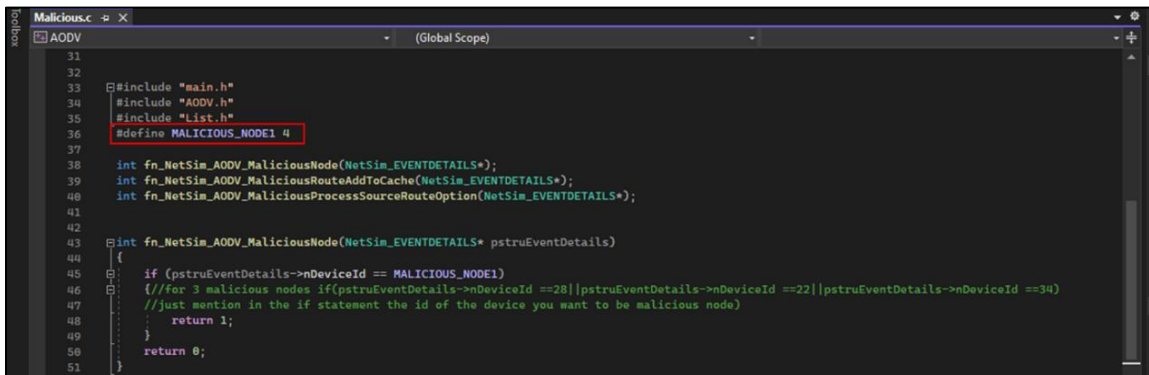A file named malicious.c is added to the AODV project, which contains the following functions:

- **fn_NetSim_AODV_MaliciousNode();** // This function is used to identify whether the current device is malicious or not in order to establish malicious behavior.

- **fn_NetSim_AODV_MaliciousRouteAddToCache();** // This function is used to add a fake route entry into the route cache of the malicious device, with its next hop as the destination.

- **fn_NetSim_AODV_MaliciousProcessSourceRouteOption();** // This function is used to drop the received packets if the device is malicious, instead of forwarding the packet to the next hop.

You can set any device as a malicious node, and you can have more than one malicious node in a scenario. The device IDs of the malicious nodes can be set inside the **fn_NetSim_AODV_MaliciousNode()** function.

Steps to simulate:

1. Open the source code in Visual studio by navigating to "Your work" on the home screen of NetSim. Then, select "Open, Reset or Compare" and click on "Open Code".

2. Expand the AODV project, open the malicious.c file, and set the malicious node id.

Figure 2: Set malicious node in the malicious .c file

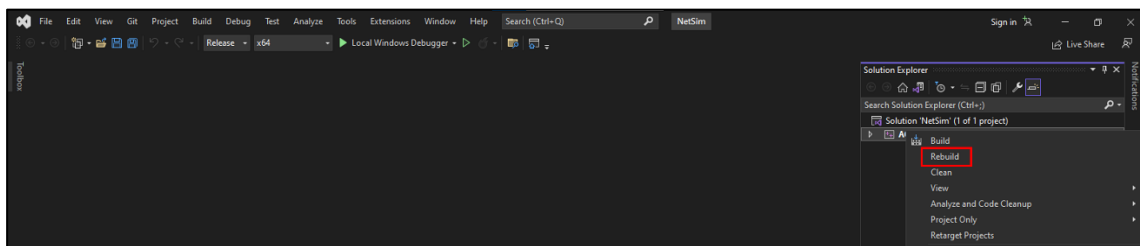3. Now right-click on the AODV project and rebuild it.

Figure 3: NetSim AODV project source code in Visual studio

4. Upon rebuilding, libAODV.dll will automatically get updated in the respective bin folder of the current workspace.

3 Example

1. The **Sinkhole-Attack-in-AODV-Workspace** comes with a sample network configuration that is already saved. To open this example, go to "Your work" in the home screen of NetSim and click on the **Sinkhole-Attack-in-AODV-Example** from the list of experiments.
2. The network consists of 7 wireless nodes with the properties configured as shown below:
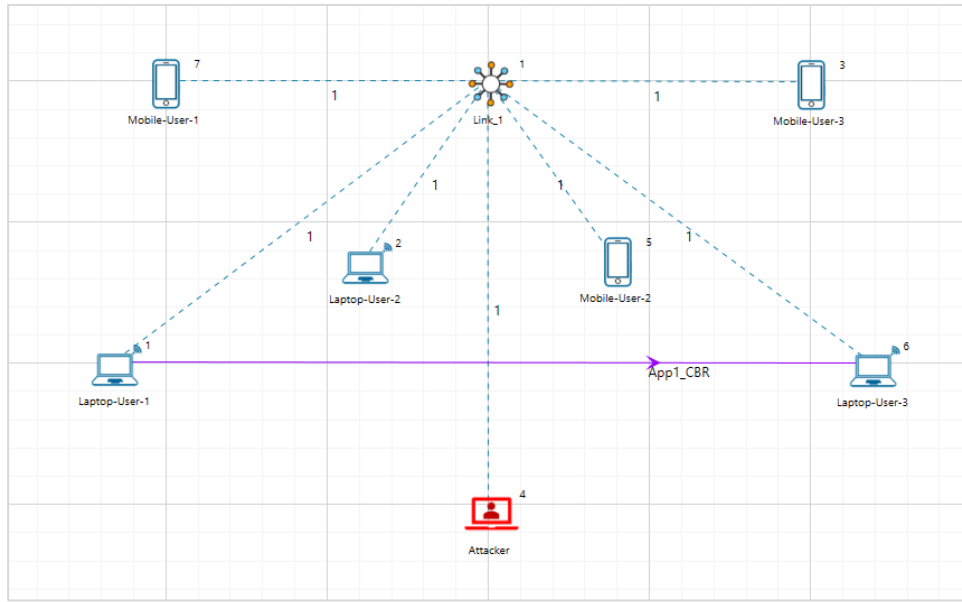
Figure 4: Network Topology

3. Set the Application properties.

Application Properties	
Source ID	1
Destination ID	6

Table 1: Application Properties

4. In the Link set the Channel characteristics: **Pathloss only**, Pathloss model: **Log Distance**, Pathloss exponent: **2.5**

5. By enabling packet trace, run the simulation for 100 seconds.

4   Results and discussion

In the packet trace, we illustrate the impact of a malicious node within the MANET, showcasing how it disrupts the normal data transfer process.

Figure 5: Analysis of results using packet trace

1. From the above screenshot, you will see that the malicious node, node-4, sends an **AODV_RREP** (Route Reply) upon receiving an **AODV_RREQ** (Route Request), attracting packets toward it.

2. While node-4 (the malicious node) tries to send a Route Reply (**AODV_RREP**), the legitimate destination, Node-6, also attempts to reply with **AODV_RREP** packets.

3. However, node-4's **AODV_RREP** reply is favoured because it pretends to be the next node to the destination, even though node-6 is the actual destination.

4. The malicious node-4 tries to mislead the network by pretending to be closer to the destination compared to the route reply sent by the actual destination.

5. You will also find that the data packets generated by node-1 are not forwarded by the malicious node-4.

This will have a direct impact on the application throughput, which can be observed in the application metrics table in the NetSim simulation results window. The throughput for application 1 registers as zero due to the presence of a malicious node (device Id 4) in the network. This node intentionally drops all data packets instead of transmitting them to their intended destination.

Figure 6: Application metrics in Results dashboard window.

5 References

[1] F. A. K. Humaira Ehsan, "Malicious AODV Implementation and Analysis of Routing Attacks in MANETs," *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications.*