Sink Hole Attack using AODV in NetSim

Software Recommended: NetSim Standard v11.0, Visual Studio 2015/2017
Project Download Link:
https://github.com/NetSim-TETCOS/SINK_HOLE_AODV_v11.0/archive/master.zip

Sinkhole attack is one of the severe attacks in wireless Ad hoc network. In sinkhole Attack, a compromised node or malicious node advertises wrong routing information to produce itself as a specific node and receives whole network traffic. After receiving 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 networks protocols such as DSR, AODV protocol.

Implementation in AODV:

- In AODV the source broadcasts RREQ packet during Route Discovery.
- The destination on receiving the RREQ packet replies with a RREP packet containing the route to reach the destination.
- But Intermediate nodes can also send RREP packet to the source if they have a route to the destination in their route cache.
- Using this as an advantage the malicious node adds a fake route entry into its route cache with the destination node as its next hop.
- On receiving the RREQ packet from the source the malicious node sends a fake RREP packet with the fake route.
- The source node on receiving this packet observes this as a better route to the destination.
- All the Network Traffic is attracted towards the Sinkhole (Malicious Node) and it can either modify the packet Information or simply drop the packet.

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

- `fn_NetSim_AODV_MaliciousNode ( )`

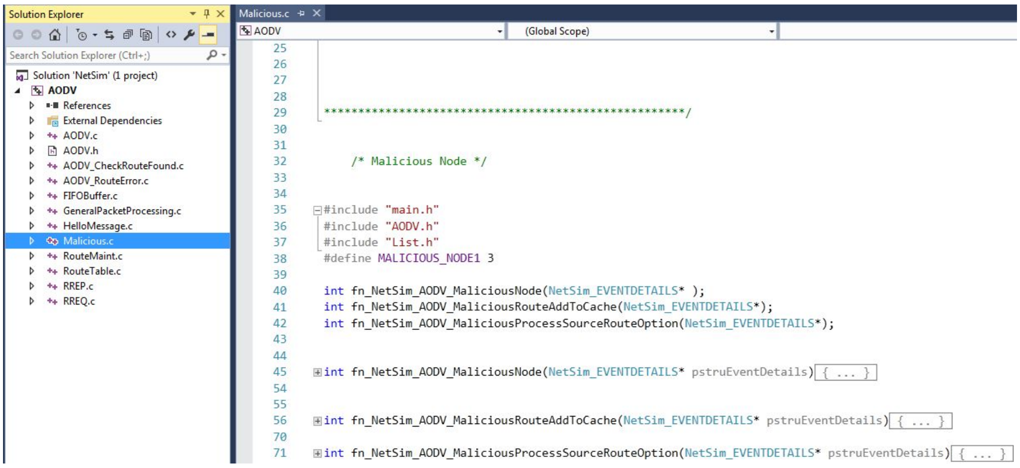
This function is used to identify whether a current device is malicious or not in-order to establish malicious behavior.

- `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.



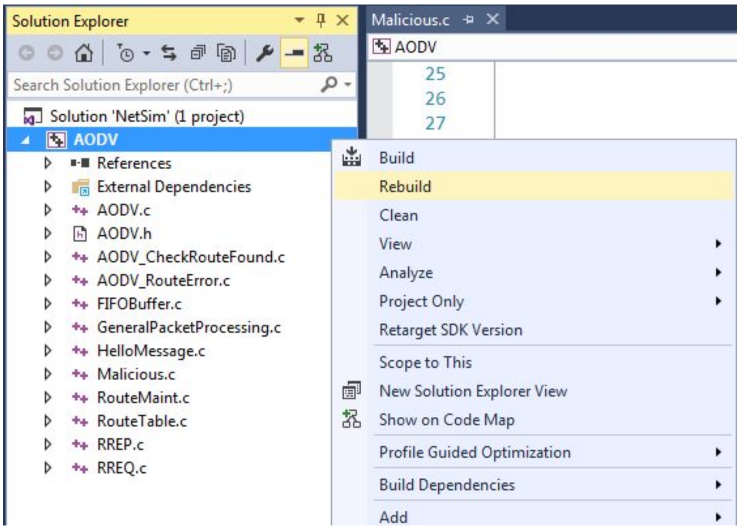
```
****************************************************/


    /* Malicious Node */


#include "main.h"
#include "AODV.h"
#include "List.h"
#define MALICIOUS_NODE1 3

int fn_NetSim_AODV_MaliciousNode(NetSim_EVENTDETAILS* );
int fn_NetSim_AODV_MaliciousRouteAddToCache(NetSim_EVENTDETAILS*);
int fn_NetSim_AODV_MaliciousProcessSourceRouteOption(NetSim_EVENTDETAILS*);


int fn_NetSim_AODV_MaliciousNode(NetSim_EVENTDETAILS* pstruEventDetails){ ... }


int fn_NetSim_AODV_MaliciousRouteAddToCache(NetSim_EVENTDETAILS* pstruEventDetails){ ... }

int fn_NetSim_AODV_MaliciousProcessSourceRouteOption(NetSim_EVENTDETAILS* pstruEventDetails){ ... }
```

You can set any device as malicious and you can have more than one malicious node in a scenario. Device id's of malicious nodes can be set inside the fn_NetSim_AODV_MaliciousNode () function.

Steps:

1. Open the Code folder and double click on the NetSim.sln fie to open the project in Visual Studio

2. Now right click on AODV project in the solution explorer and select Rebuild.

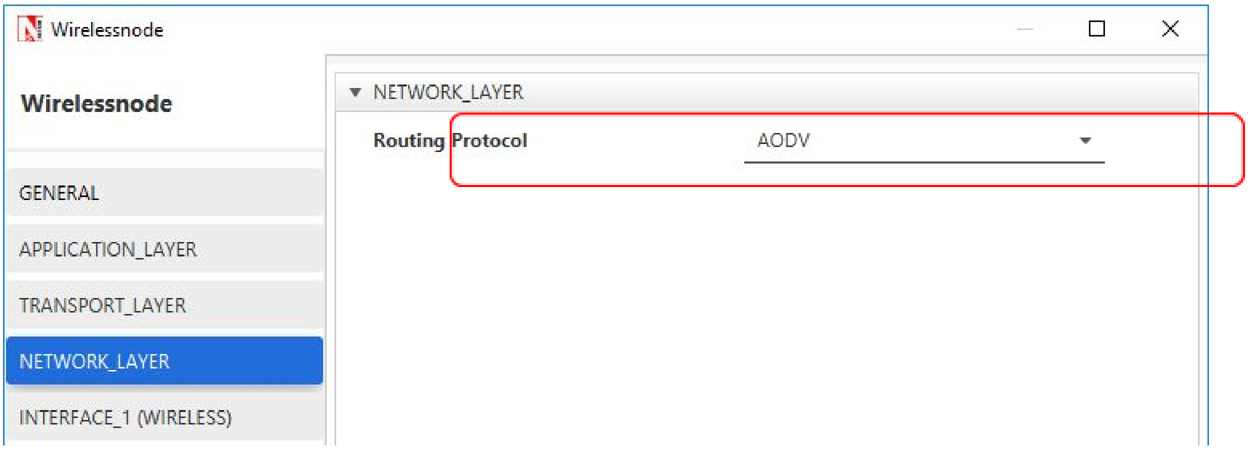


3. Now Copy the newly built libAODV.dll from the DLL folder inside the Simulation – Sinkhole Attack directory.

4. Replace the DLL in NetSim bin folder in the NetSim installation directory, after renaming the original libAODV.dll file.

5. Create a network scenario in MANET with UDP running in the Transport Layer.

6. Change the Network Layer Routing protocol to **AODV**.



Wirelessnode

Wirelessnode

GENERAL

APPLICATION_LAYER

TRANSPORT_LAYER

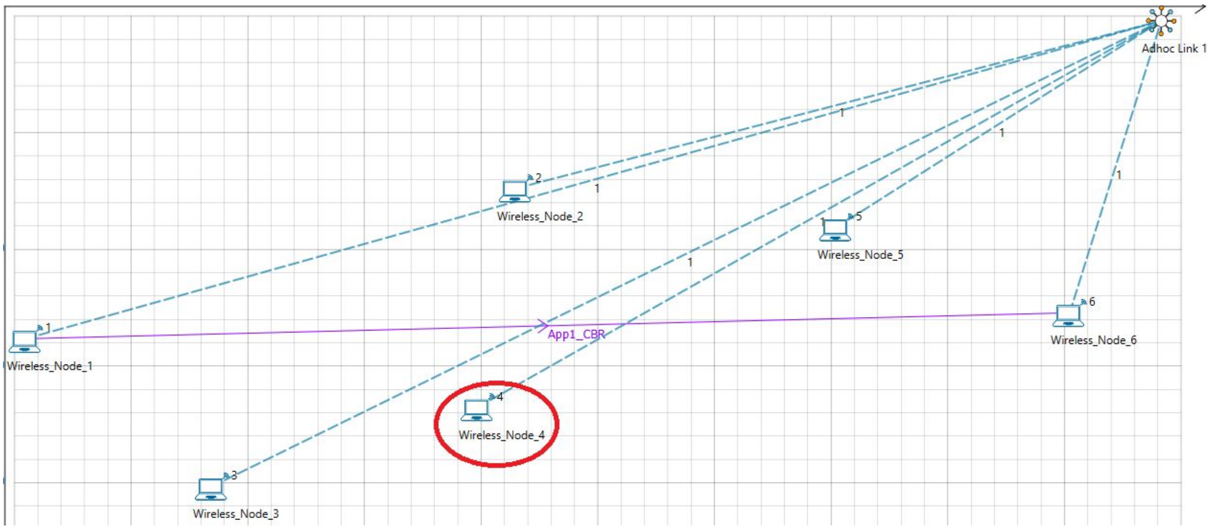
NETWORK_LAYER

INTERFACE_1 (WIRELESS)

NETWORK_LAYER

Routing Protocol AODV

7. For example, you can create a scenario as shown in the following screenshot:



Scenario Steps: (Wireless Node 6)

Source – Device id 1
Destination – Device id 6
Sinkhole (Malicious node) – Device id 4

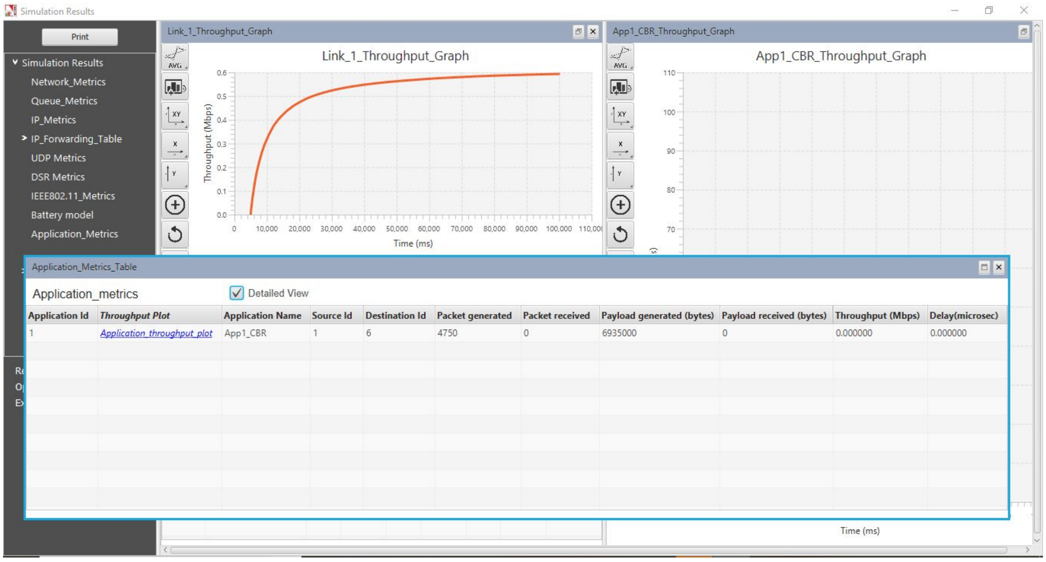
Link properties (Adhoc Link1)

Channel characteristics – Pathloss only
Path Loss model – LOG DISTANCE
Path Loss Exponent: 3

8. Run the Simulation for 100 seconds.

9. View the packet animation. You will find that the malicious node (Device id 4) gives Route Reply on receiving Route Request and attracts packets towards it. You will also find that the malicious node does not forward the packets that it receives.

10. This will have a direct impact on the Application Throughput which can be observed in the Application Metrics table present in NetSim Simulation Results window.



Application_metrics ☑ Detailed View

Application Id	*Throughput Plot*	Application Name	Source Id	Destination Id	Packet generated	Packet received	Payload generated (bytes)	Payload received (bytes)	Throughput (Mbps)	Delay(microsec)
1	*Application_throughput_plot*	App1_CBR	1	6	4750	0	6935000	0	0.000000	0.000000