

# Sink Hole Attack in MANET using DSR

---

**Software Recommended:** NetSim Standard v13.0, Visual Studio 2017/2019

**Project Download Link:**

[https://github.com/NetSim-TETCOS/SINK\\_HOLE\\_ATTACK\\_AODV\\_v13\\_0/archive/refs/heads/main.zip](https://github.com/NetSim-TETCOS/SINK_HOLE_ATTACK_AODV_v13_0/archive/refs/heads/main.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 protocol.

**Implementation in DSR:**

- In DSR 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.

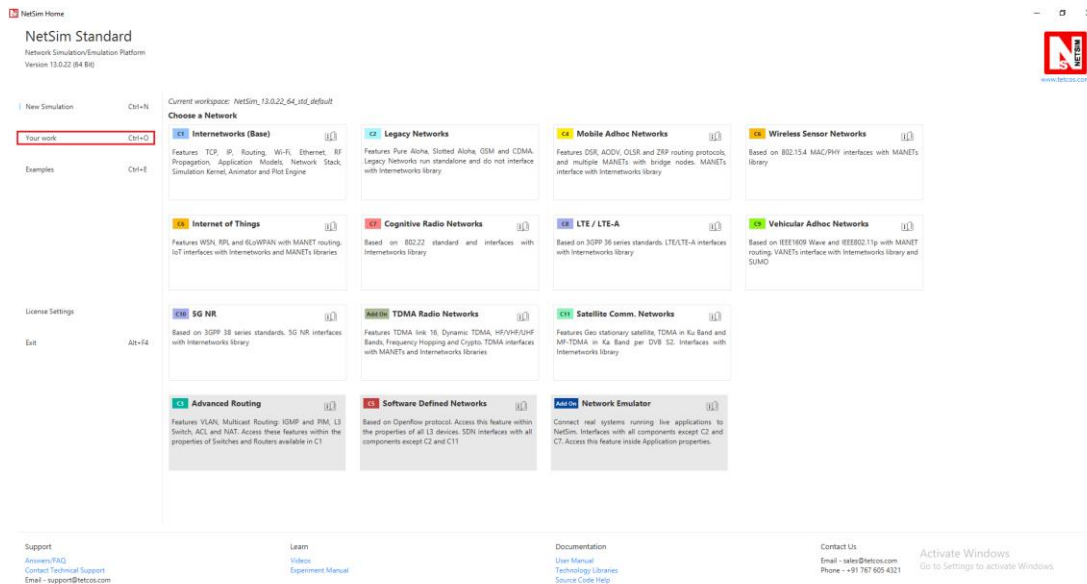
A file **malicious.c** is added to the DSR project which contains the following functions:

- **fn\_NetSim\_DSR\_MaliciousNode()**  
This function is used to identify whether a current device is malicious or not in-order to establish malicious behavior.
- **fn\_NetSim\_DSR\_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\_DSR\_MaliciousProcessSourceRouteOption()**  
This function is used to drop the received packets if the device is malicious, instead of forwarding the packet to the next hop.

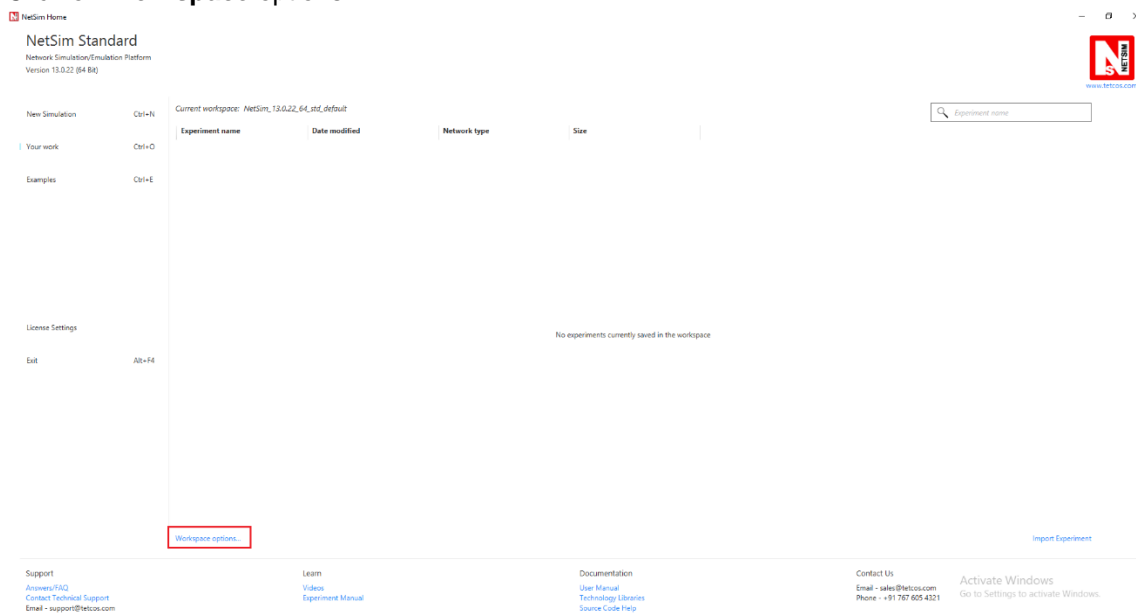
You can set any device as malicious and you can have more than one malicious node in a scenario. Device id's of malicious nodes can be set inside the **fn\_NetSim\_DSR\_MaliciousNode()** function.

**Steps:**

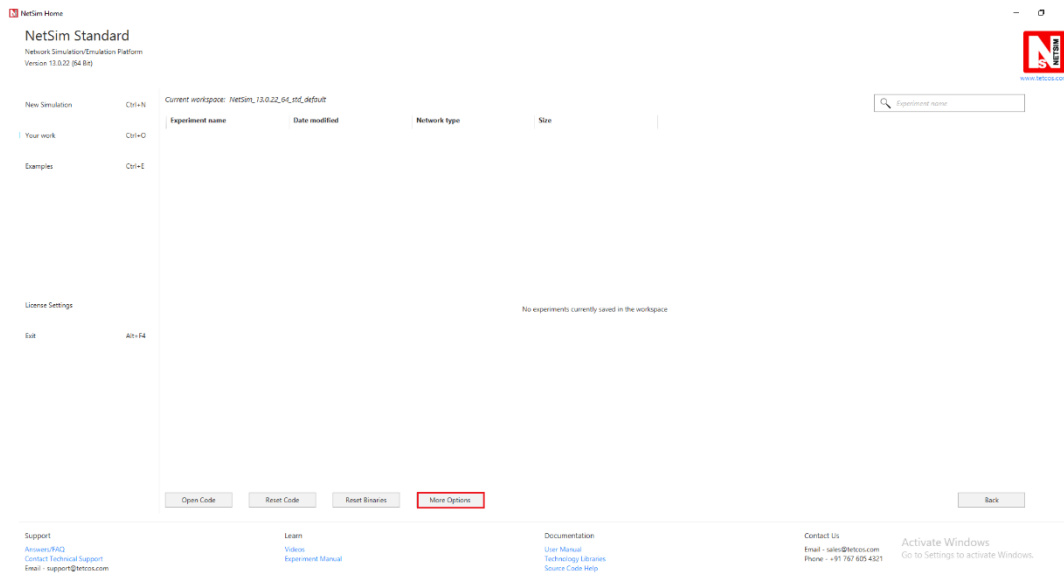
1. After you unzip the downloaded project folder, Open NetSim Home Page click on **Your work** option,



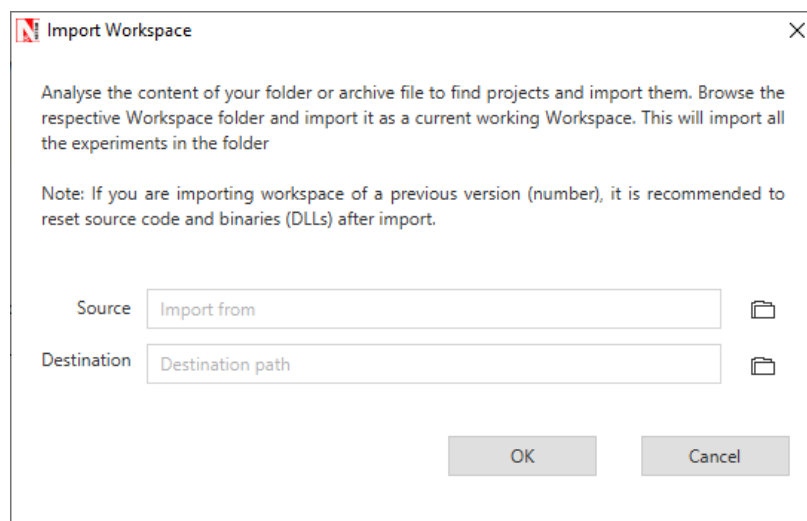
## 2. Click on **Workspace** options



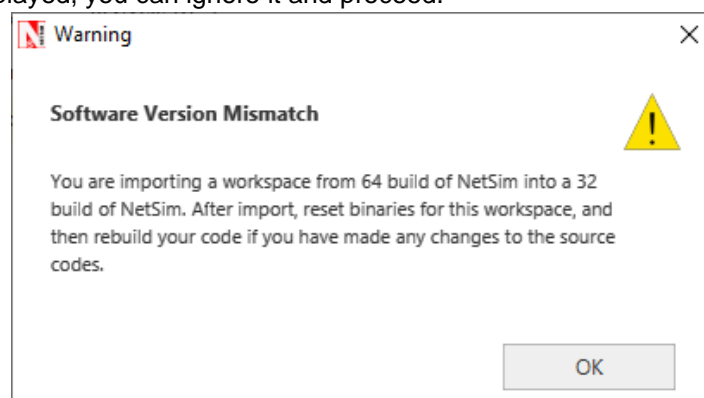
## 3. Click on **More Options**,



4. Go to Your work->Workspace Options->More Options in NetSim Home window. Click on the Import button. For the source path option browse and select the SINK\_HOLE\_ATTACK\_DSR\_WorkSpace.netsim\_wsp file from the extracted project folder. Also set the destination path where you want to setup the workspace. Click on OK.

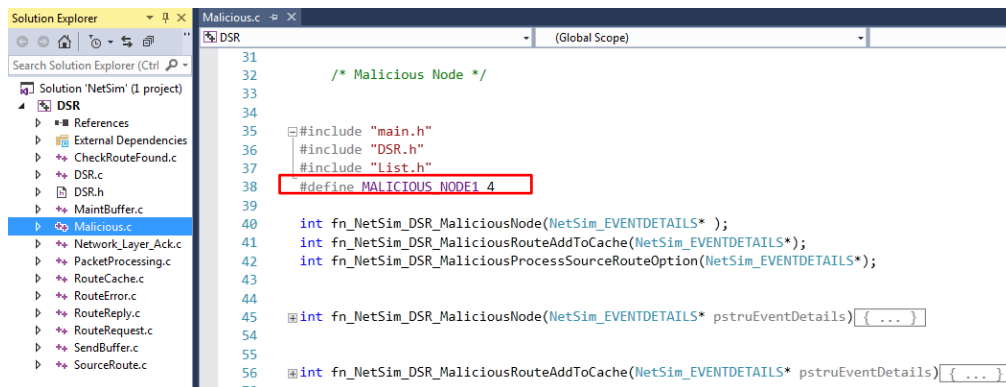


5. While importing the workspace, if the following warning message indicating Software Version Mismatch is displayed, you can ignore it and proceed.



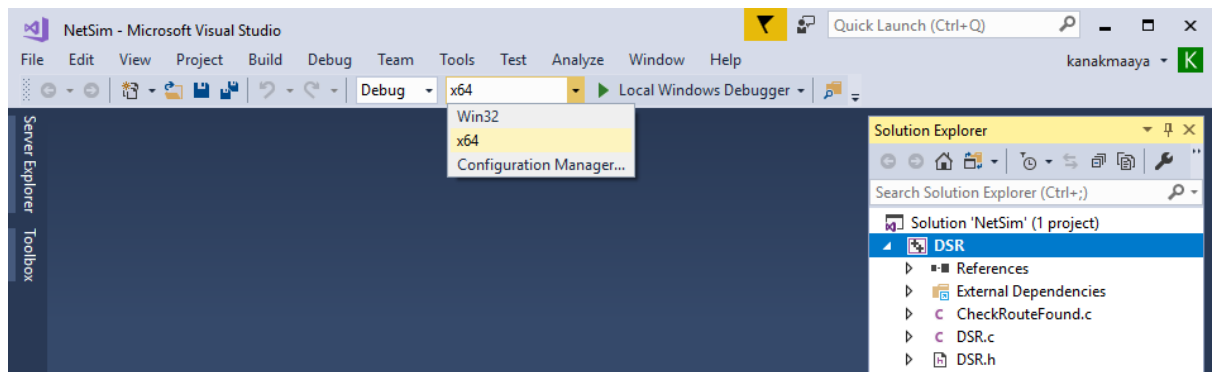
6. Go to home page, Click on Your work → Workspace options → Open code

1. Expand DSR project and open Malicious.c file.
2. Set malicious node id.

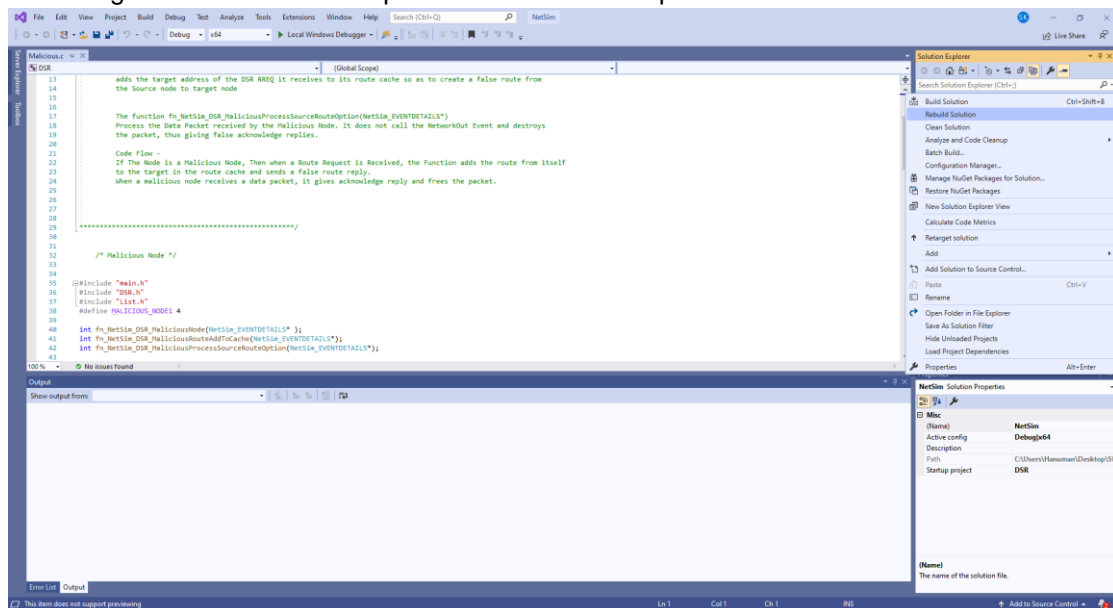


```
31
32      /* Malicious Node */
33
34
35      #include "main.h"
36      #include "DSR.h"
37      #include "List.h"
38      #define MALICIOUS_NODE1 4
39
40      int fn_NetSim_DSR_MaliciousNode(NetSim_EVENTDETAILS* );
41      int fn_NetSim_DSR_MaliciousRouteAddToCache(NetSim_EVENTDETAILS*);
42      int fn_NetSim_DSR_MaliciousProcessSourceRouteOption(NetSim_EVENTDETAILS*);
43
44
45      int fn_NetSim_DSR_MaliciousNode(NetSim_EVENTDETAILS* pstruEventDetails){...}
46
47
48
49
50
51
52
53
54
55
56      int fn_NetSim_DSR_MaliciousRouteAddToCache(NetSim_EVENTDETAILS* pstruEventDetails){...}
57
58
59
60
61
62
63
64
65
66
67
68
69
70
```

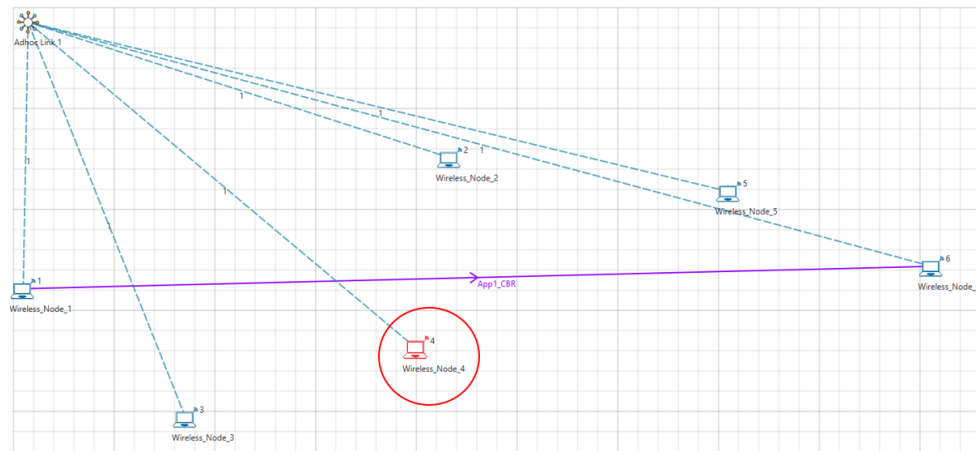
1. Based on whether you are using NetSim 32 bit or 64 bit setup you can configure Visual studio to build 32 bit or 64 bit DLL files respectively as shown below:



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



3. Upon rebuilding, libDSR.dll will automatically get updated in the respective bin folder of the current workspace.
4. Create a network scenario in MANET with UDP running in the Transport Layer.
5. For example, you can create a scenario as shown in the following screenshot:



Source – Device id 1

Destination – Device id 6

Sinkhole (malicious node) – Device id 4

### Link Properties (Adhoc link1)

Channel characteristics – Path Loss only

Path Loss model – LOG DISTANCE

Path Loss Exponent: 3

6. Run the Simulation for 100 seconds.
7. 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.
8. 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.

Simulation Results

Network Performance

Link\_Metrics

Queue\_Metrics

TCP\_Metrics

IP\_Metrics

IP\_Forwarding\_Table

UDP\_Metrics

DOR\_Metrics

IEEE802.11\_Metrics

Battery model

Application\_Metrics

Export Results (.xml/.csv)

Print Results (HTML)

Open Packet Trace

Open Event Trace

Log Files

Reverts To Original View

Application\_Metrics\_Table

Application Id

Application Name

Packet generated

Packet received

Throughput (Mbps)

Delay (microsec)

Jitter (microsec)

1	App1_CBR	4750	0	0.000000	0.000000	0.000000
---	----------	------	---	----------	----------	----------

Link\_Metrics\_Table

Link\_id

Link\_throughput\_gbit

Packet\_transmits...

Packet\_errored...

Packet\_collided...

All	NA	4777	4770	31	0	0	0
1	NA	4777	4770	31	0	0	0

TCP\_Metrics\_Table

Source

Destination

Segment Sent

Segment Received

Ack Sent

Ack Received

Duplicate ack received

WIRELESS_NODE_1	ANY_DEVICE	0	0	0	0	0
WIRELESS_NODE_2	ANY_DEVICE	0	0	0	0	0
WIRELESS_NODE_3	ANY_DEVICE	0	0	0	0	0
WIRELESS_NODE_4	ANY_DEVICE	0	0	0	0	0
WIRELESS_NODE_5	ANY_DEVICE	0	0	0	0	0
WIRELESS_NODE_6	ANY_DEVICE	0	0	0	0	0

Queue\_Metrics\_Table

Device\_id

PortId

Queueved\_pkts...

Dequeueved\_pkts...

Dropped\_pkts...

No content in table

Activate Windows  
Go to Settings to activate Windows.