Sink Hole Attack in AODV

Software Recommended: NetSim Standard v12.0, Visual Studio 2017/2019

Follow the instructions specified in the following link to clone/download the project folder from GitHub using Visual Studio:
https://tetcos.freshdesk.com/support/solutions/articles/14000099351-how-to-clone-netsim-file-exchange-project-repositories-from-github-

Other tools such as GitHub Desktop, SVN Client, Sourcetree, Git from the command line, or any client you like to clone the Git repository.

Note: It is recommended not to download the project as an archive (compressed zip) to avoid incompatibility while importing workspaces into NetSim.

Secure URL for the GitHub repository:

https://github.com/NetSim-TETCOS/SINK_HOLE_ATTACK_AODV_v12.0.git

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.

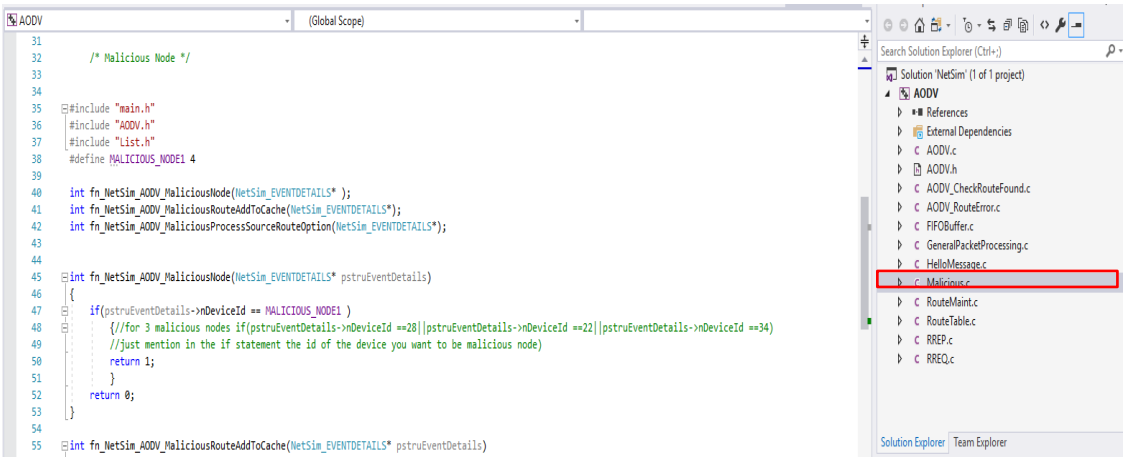
A file **malicious.c** is added to the AODV project which contains the following functions:

- `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 ()`

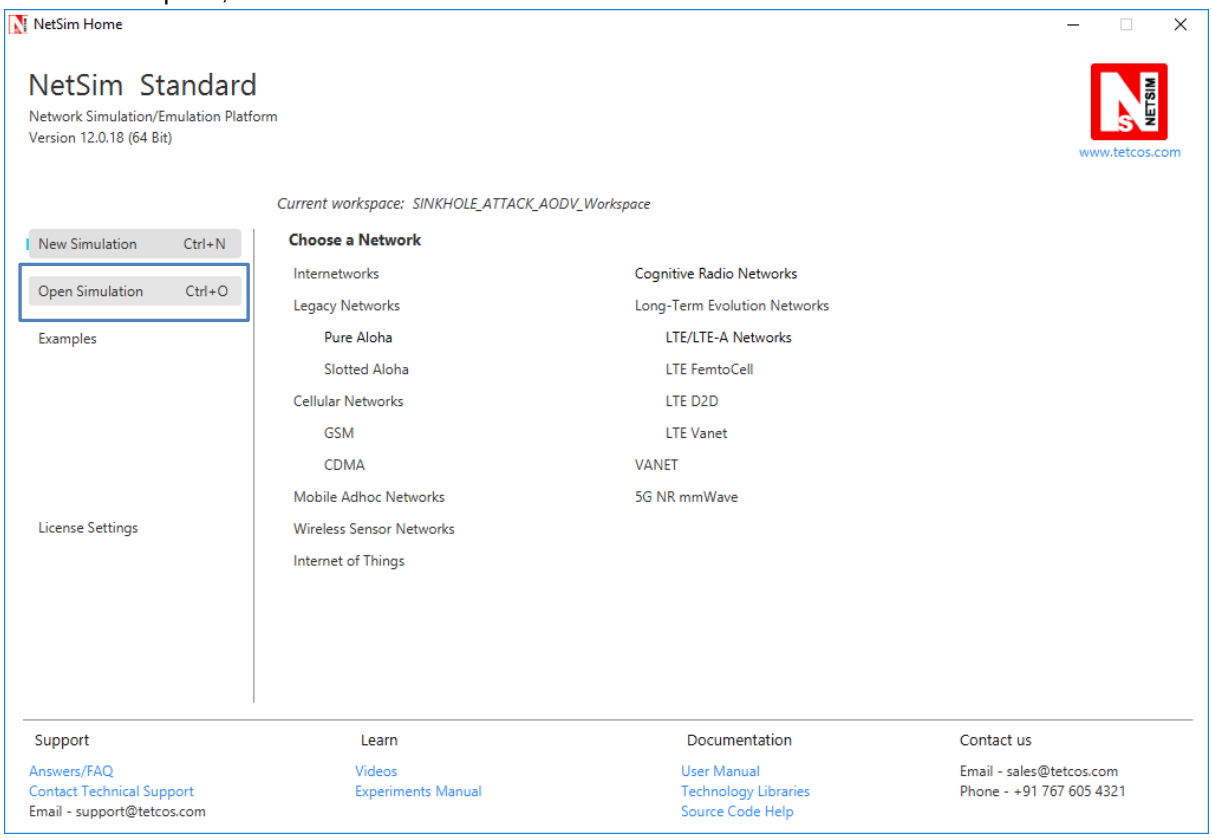
This function is used to drop the received packets if the device is malicious, instead of forwarding the packet to the next hop



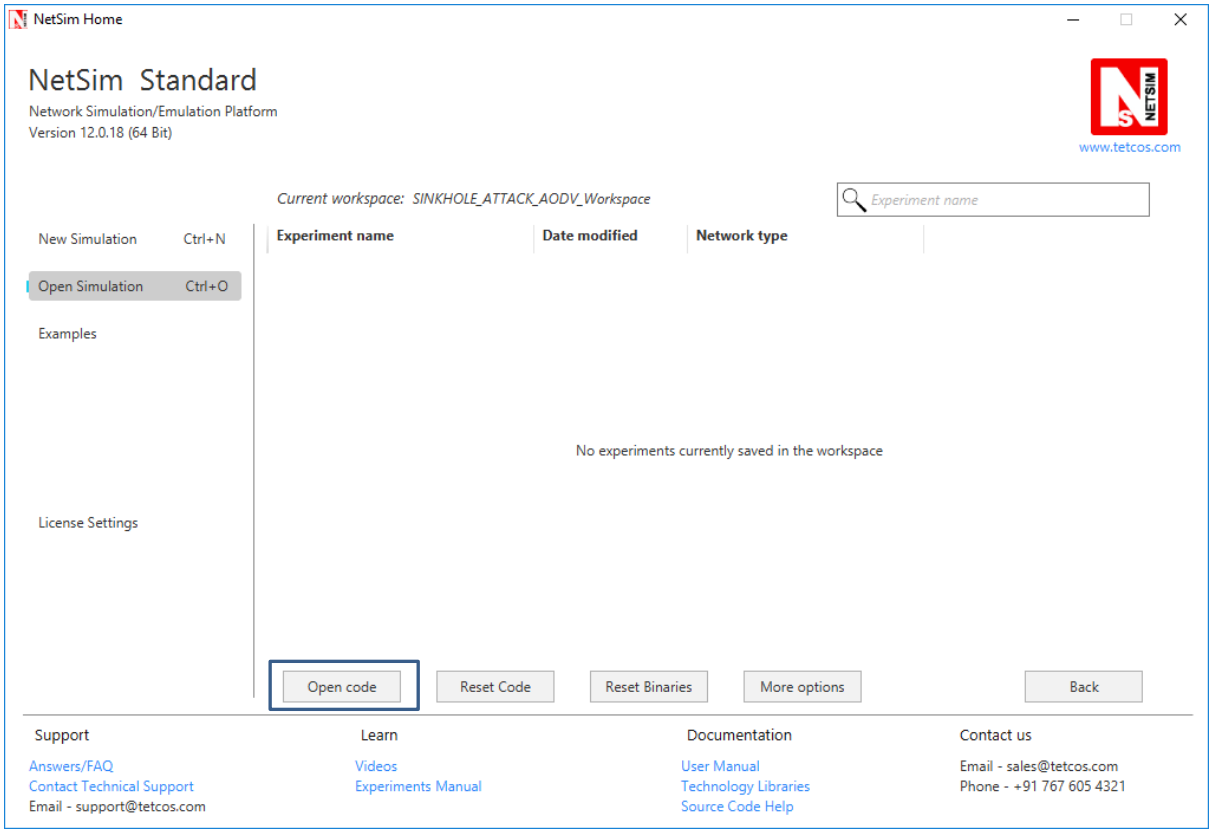
You can set any device as malicious and you can have more than one malicious node in a scenario. Device id's of malicious nodes can be set inside the fn_NetSim_AODV_MaliciousNode () function.

Steps:

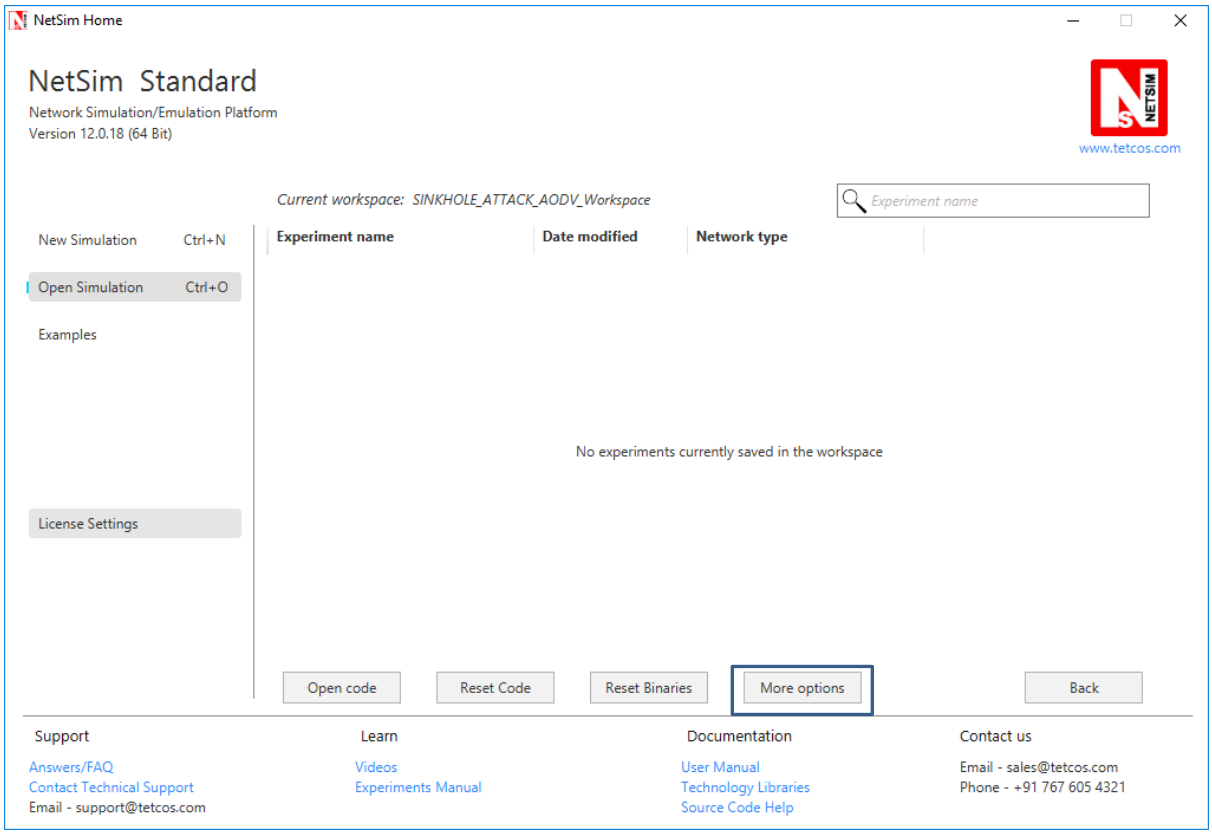
1. After you unzip the downloaded project folder, Open NetSim Home Page click on **Open Simulation** option,



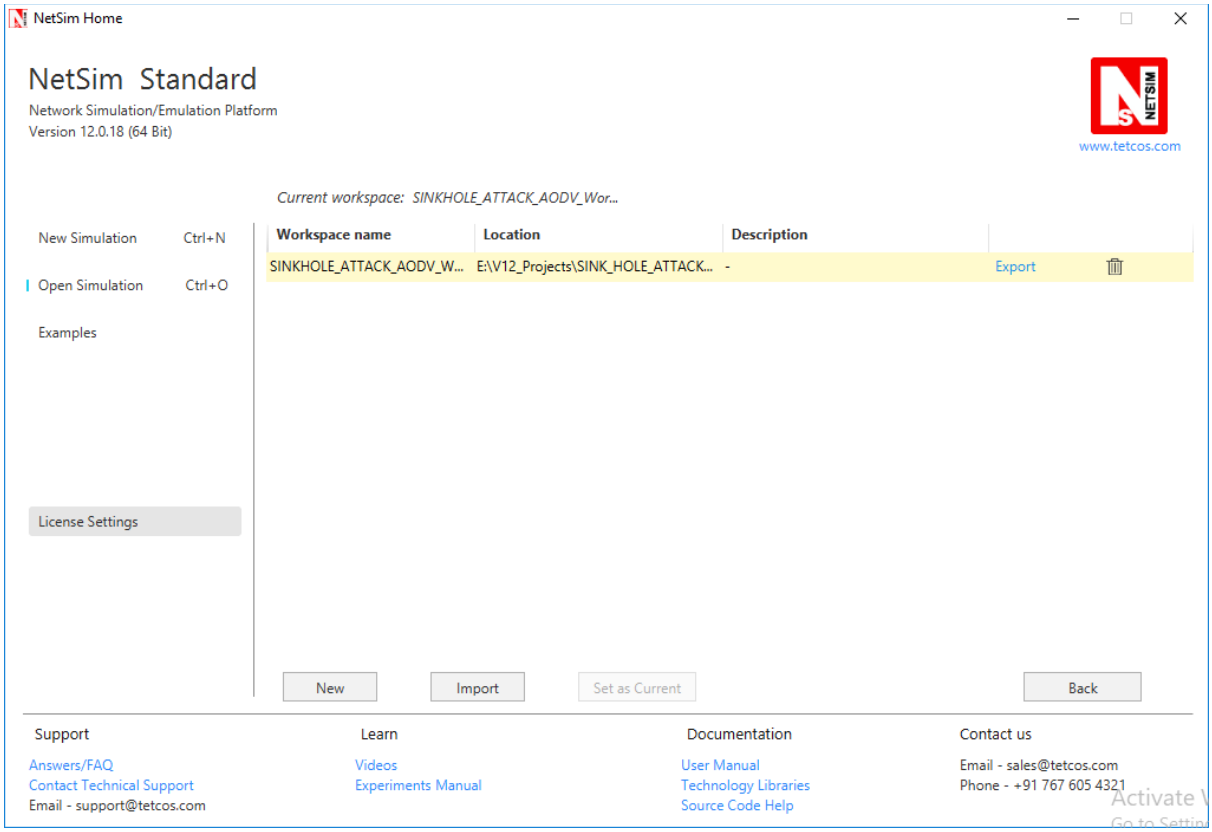
2. Click on Workspace options



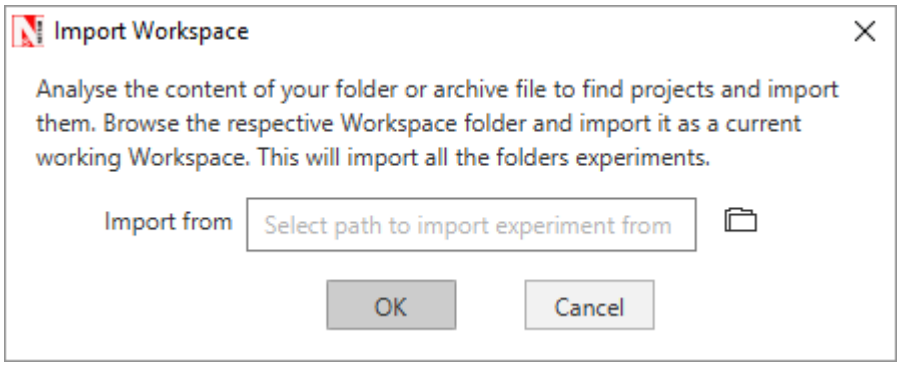
3. Click on More Options,



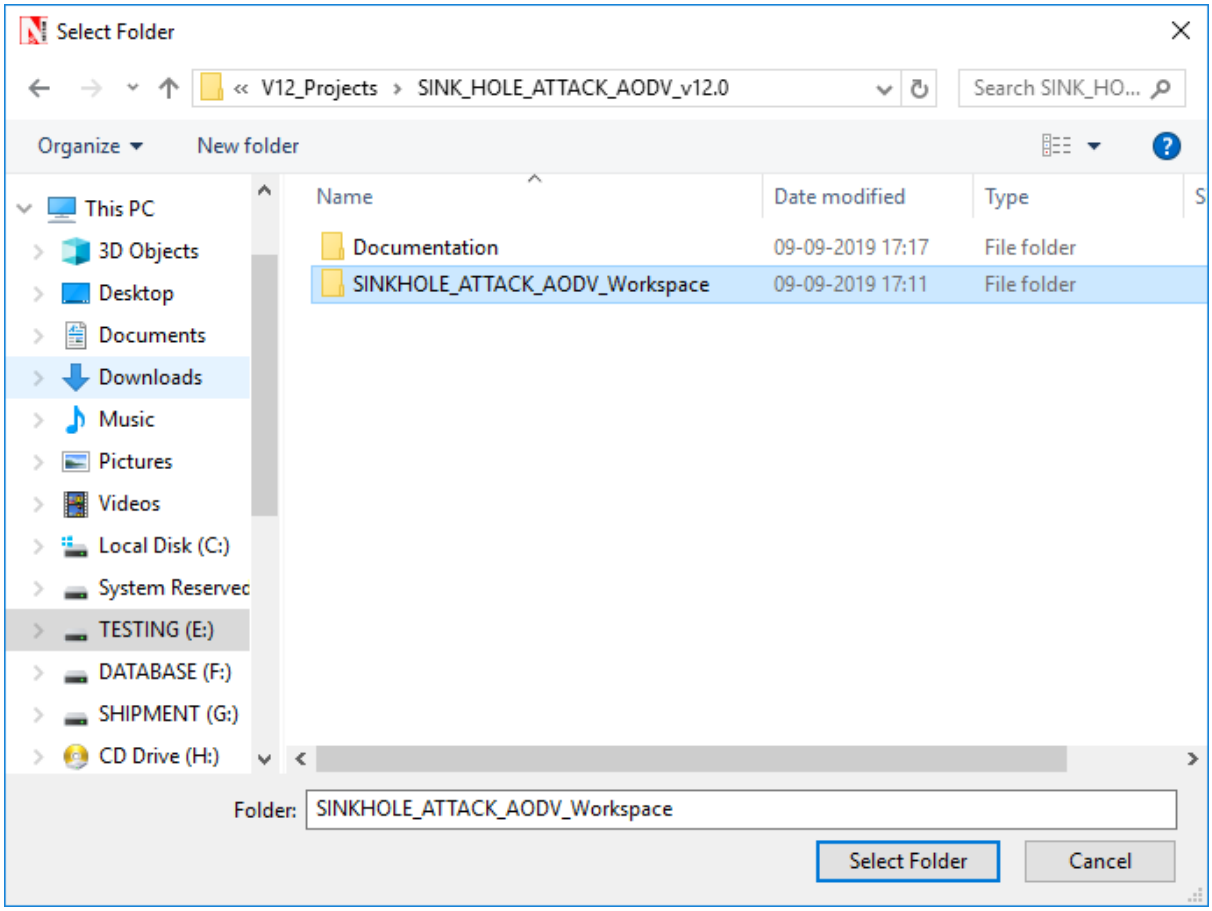
4. Click on Import option.



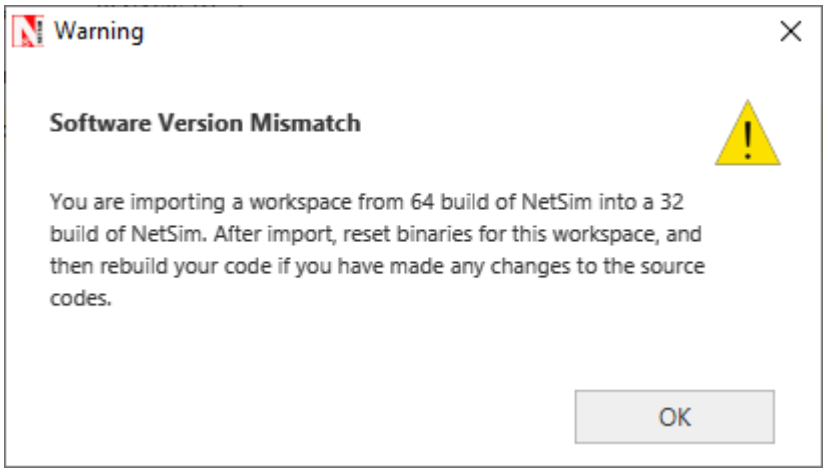
5. It displays a window where users need to give the path of the workspace folder and click on OK as shown below:



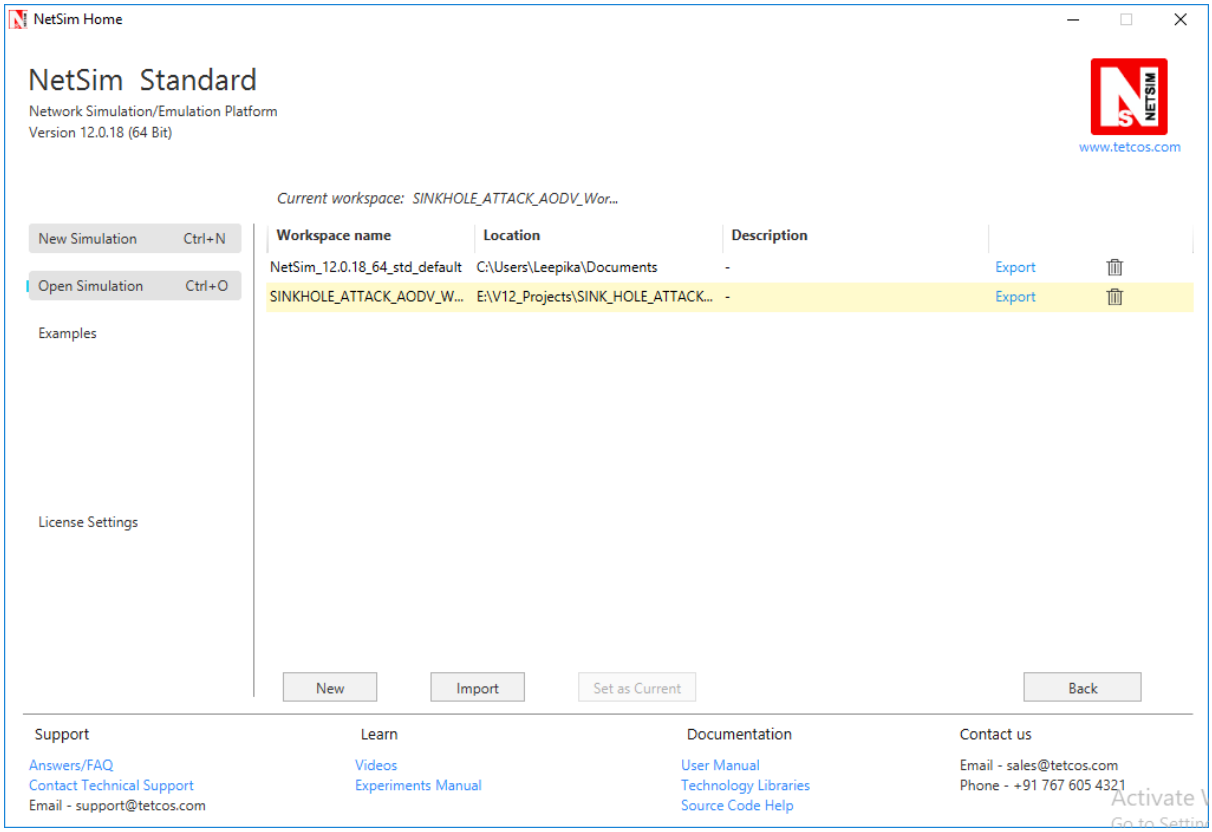
6. Browse to the Sink_Hole_Attack_AODV_Workspace folder and click on select folder as shown below:



7. After this click on OK button in the Import Workspace window.
8. While importing the workspace, if the following warning message indicating Software Version Mismatch is displayed, you can ignore it and proceed.



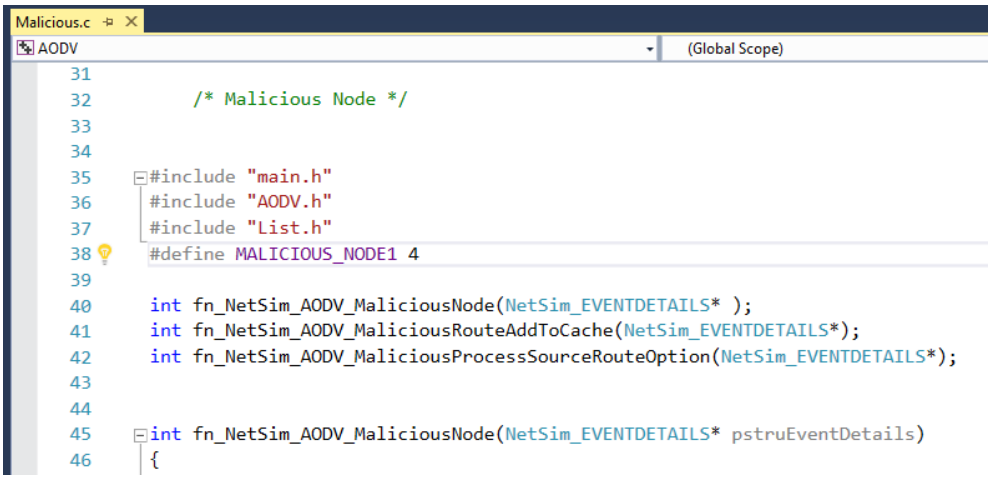
9. The Imported workspace will be set as the current workspace automatically. To see the imported workspace, click on Open Simulation->Workspace Options->More Options as shown below:



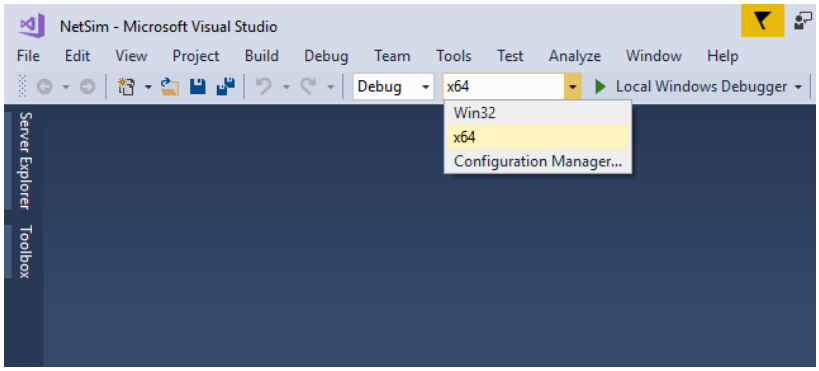
10. Go to home page, Click on Open Simulation → Workspace options → Open code

1. Expand AODV project and open Malicious.c file.

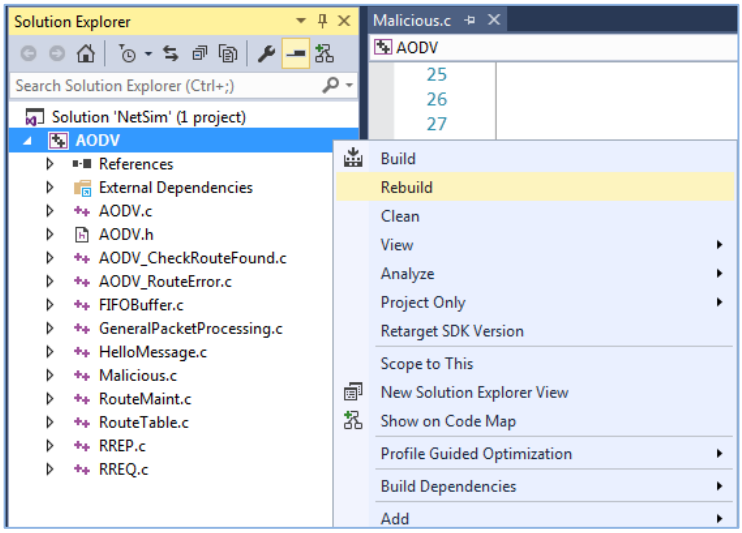
2. Set malicious node id.



3. 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:

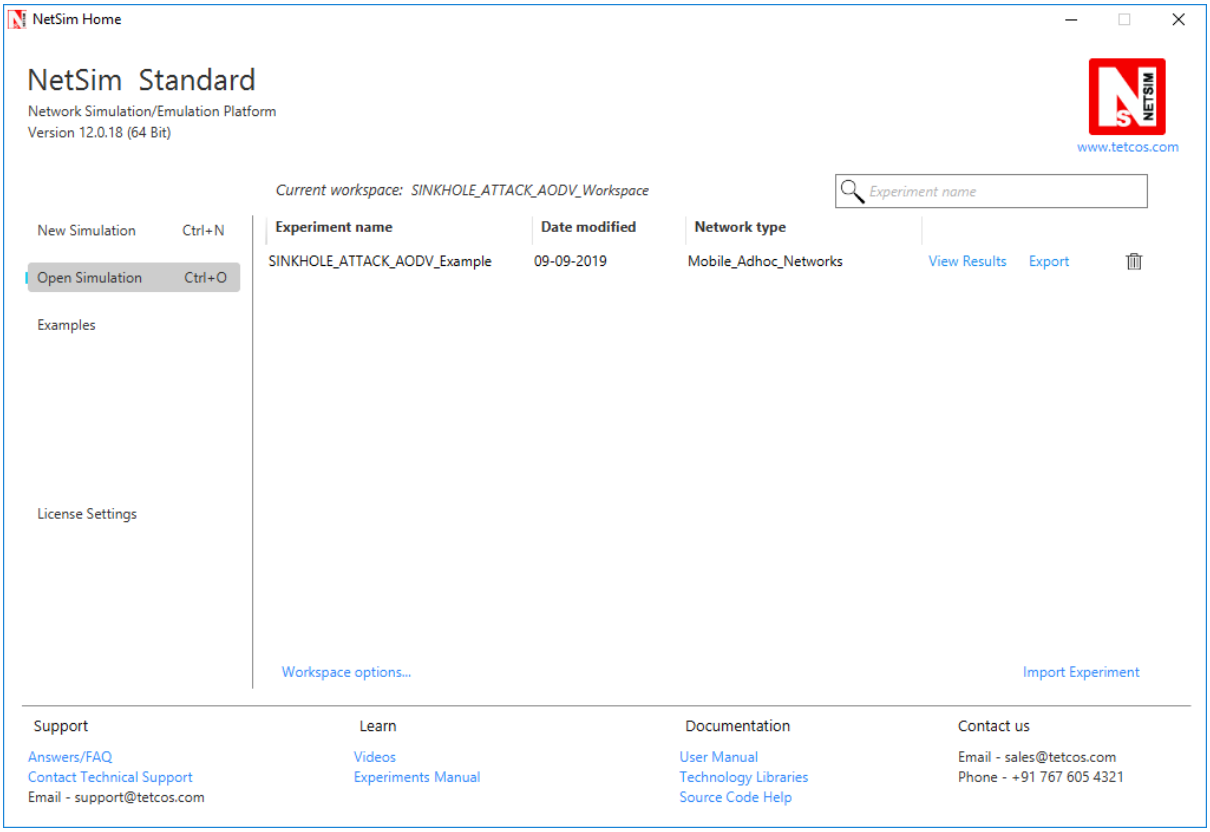


4. Now right click on AODV project in the solution explorer and select Rebuild.



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

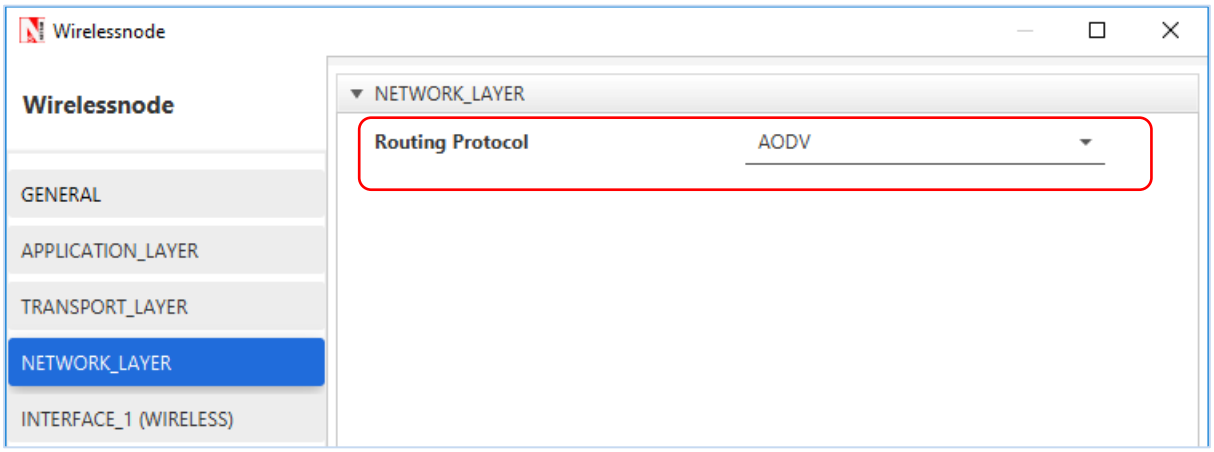
6. Then Sink_Hole_Attack_AODV_Workspace comes with a sample configuration that is already saved. To open this example, go to Open Simulation and click on Experiment that is present under the list of experiments as shown below:



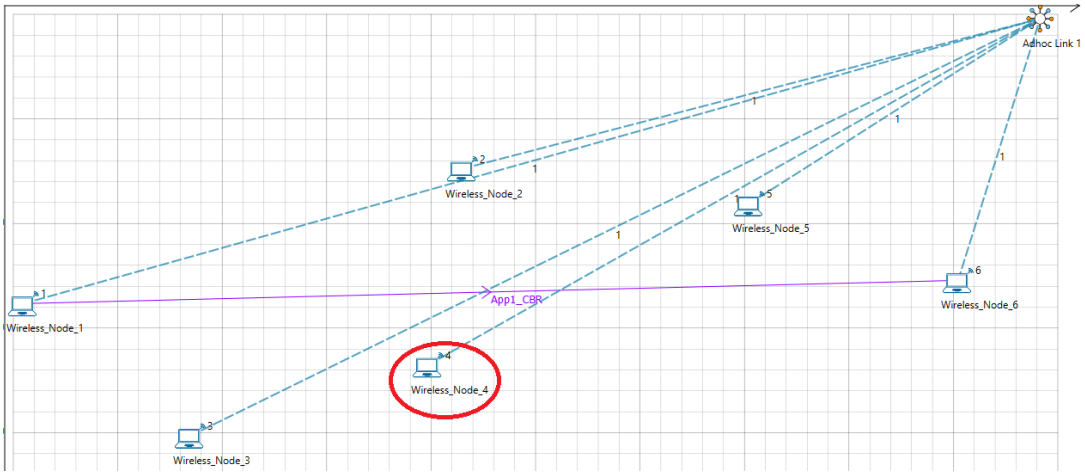
7. The following steps illustrates the creation of the above sample configuration file:

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

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



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



Scenario Steps: (Wireless Node 6)

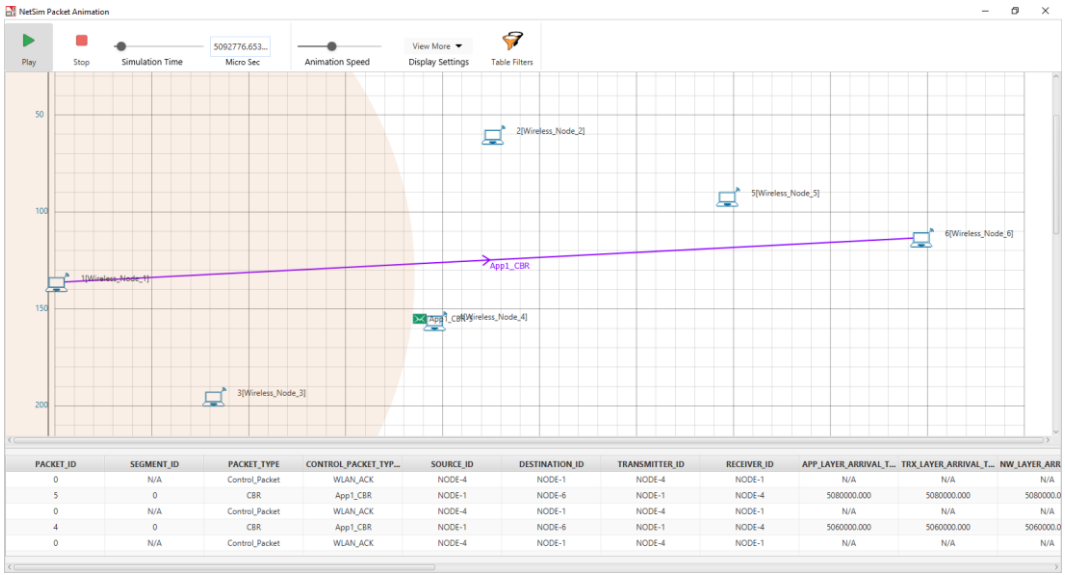
Source – Device id 1
Destination – Device id 6
Sinkhole (Malicious node) – Device id 4

Link properties (Adhoc Link1)

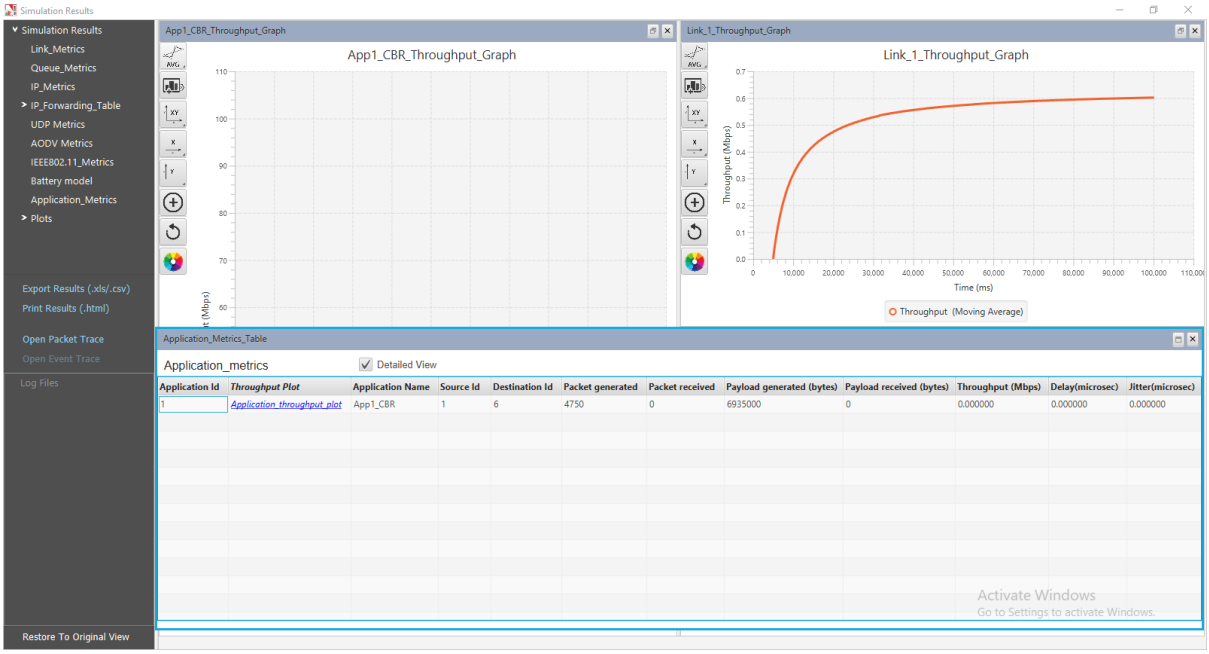
Channel characteristics – Pathloss only
Path Loss model – LOG DISTANCE
Path Loss Exponent: 3

11. Run the Simulation for 100 seconds.

12. 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.



13. 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
Simulation Results
Link_Metrics
Queue_Metrics
IP_Metrics
IP_Forwarding_Table
UDP Metrics
AODV Metrics
IEEE802.11_Metrics
Battery model
Application_Metrics
Plots
Export Results (.xls/.csv)
Print Results (.html)
Open Packet Trace
Open Event Trace
Log Files
Restore To Original View
App1_CBR_Throughput_Graph
App1_CBR_Throughput_Graph
Link_1_Throughput_Graph
Link_1_Throughput_Graph
Throughput (Mbps)
Time (ms)
Throughput (Moving Average)
Application_Metrics_Table
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)
Jitter(microsec)
1
Application_throughput_plot
App1_CBR
1
6
4750
0
6935000
0
0.000000
0.000000
0.000000
Activate Windows
Go to Settings to activate Windows.