**Sink Hole Attack using RPL in IOT**

**Software Recommended:** NetSim Standard v10.2 (64-bit), Visual Studio 2015

In sinkhole Attack, a compromised node or malicious node advertises fake rank information to form the fake routes. After receiving the message packet it drop the packet information. Sinkhole attacks affect the performance of IoT networks protocols such as RPL protocol.

**Implementation in RPL (for 1 sink)**

* In RPL the transmitter broadcasts the DIO during DODAG formation.
* The receiver on receiving the DIO from the transmitter updates its parent list, sibling list, rank and sends a DAO message with route information.
* Malicious node upon receiving the DIO message it does not update the rank instead it always advertises a fake rank.
* The other node on listening to the malicious node DIO message the update their rank according to the fake rank.
* After the formation of DODAG, if the node that is transmitting the packet has malicious node as the preferred parent, transmits the packet to it but the malicious node instead of transmitting the packet to its parent, it simply drops the packet resulting in zero throughput.

A file Malicious.c is added to the RPL project.  
The file contains the following functions

1. **fn\_NetSim\_RPL\_MaliciousNode( )**

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

1. **fn\_NetSim\_RPL\_MaliciousRank( )**

This function is used to give a fake rank to the malicious node.

1. **rpl\_drop\_msg( )**

This function is used to drop the packet by the malicious node if it enters into its network layer.

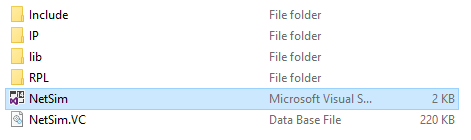
**Sink Hole attack** – The malicious node advertises the fake rank.  
**fn\_NetSim\_RPL\_MaliciousRank( )** is the sink hole attack function.

**Black Hole attack** – The malicious node drops the packet.  
**rpl\_drp\_msg()** is the black hole attack function

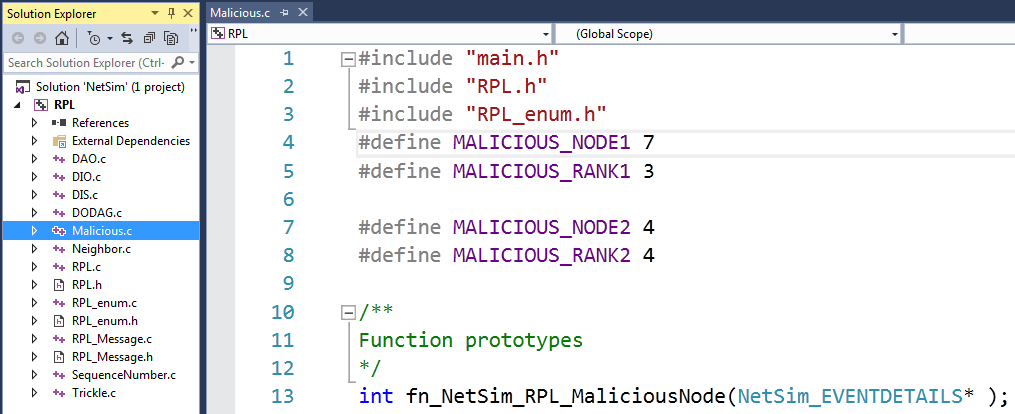
You can set any device as malicious and you can have more than one malicious node in a scenario. Device id’s of malicious nodes can be set inside the fn\_NetSim\_RPL\_MaliciousNode() function.

**Steps:**

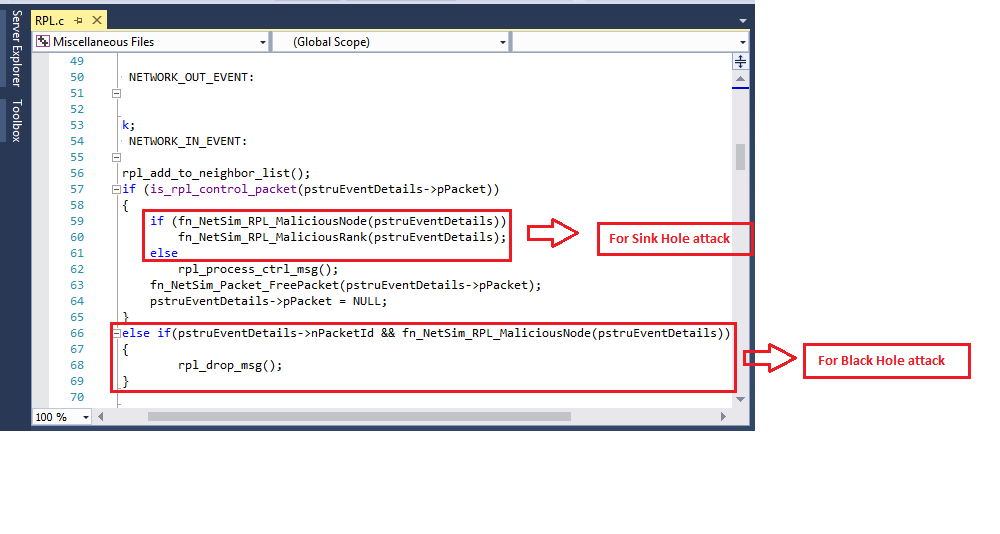
1. Open the Code folder and double click on the NetSim.sln fie to open the project in Visual Studio 2015**.**



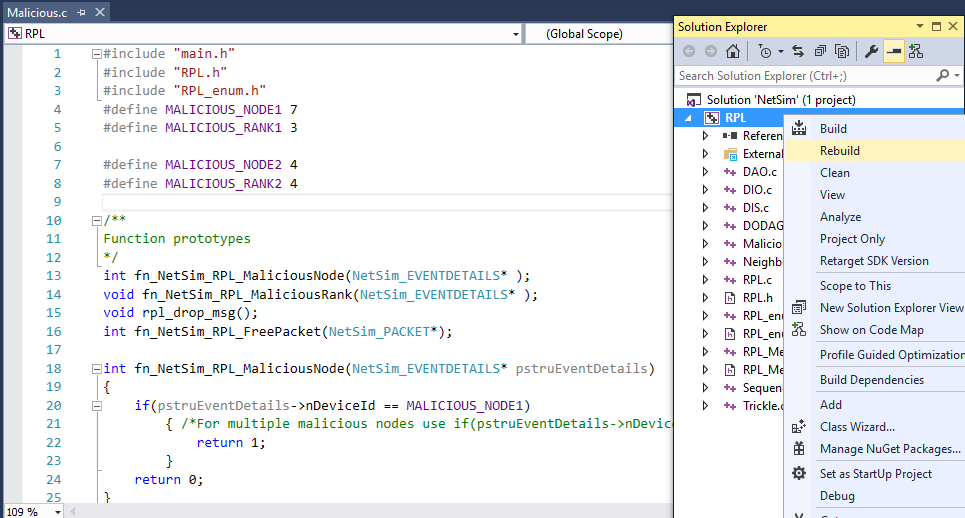
1. Set malicious node id and the fake rank.



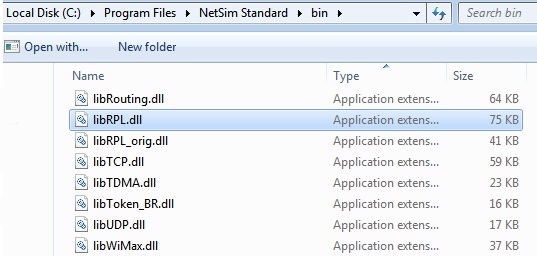
1. Add the code that is highlighted in RPL.c file



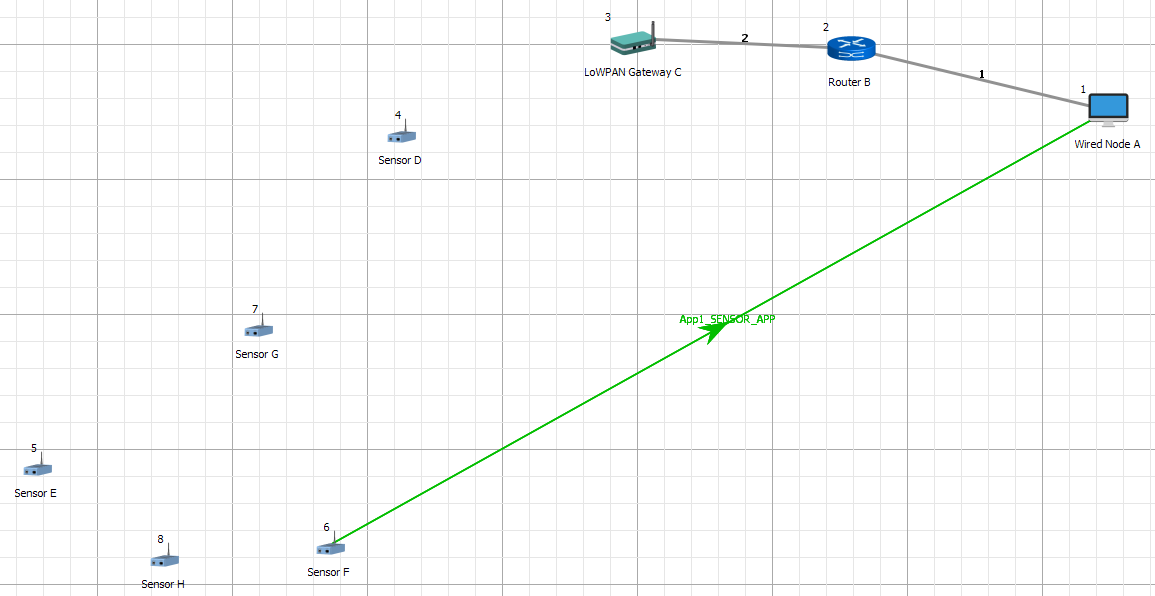
1. Now right click on RPL project in the solution explorer and select Rebuild.

****

1. Now Copy the newly built libRPL.dll from the DLL folder inside the Simulation – Sinkhole Attack directory.
2. Replace the DLL in NetSim bin folder in the NetSim installation directory, after renaming the original libRPL.dll file.

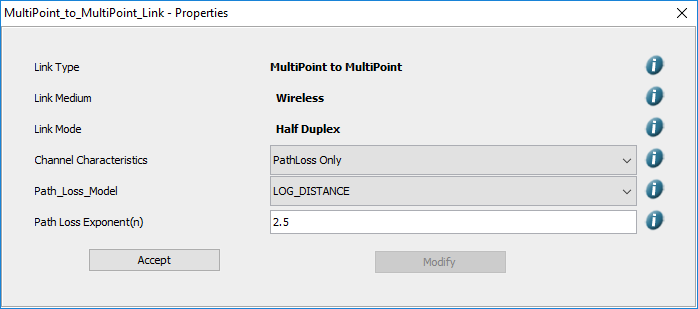
****

1. Create a network scenario in IoT with UDP running in the Transport Layer and RPL in Network Layer.
2. For example, you can create a scenario as shown in the following screenshot:



Environment Properties:

* Right click anywhere on the Environment Grid and select Properties.
* Select the Channel Characteristics and set the parameters accordingly.



C:\Users\UPAL\Desktop\winkey1.jpg**Output**

* Press + R and type %temp%, Temp folder will be opened.
* In Temp folder you will find a folder named NetSim.
* In NetSim, you will find a txt file named rpllog.txt

Open rpllog.txt then you will find the information given below

The DODAG formed in this scenario is:

(Note: For every DODAG, 6LoWPAN Gateway is the root of the DODAG.)

Root is 3 with rank = 1 (Since the Node 3 is 6LoWPAN Gateway)

Node 4(Malicious Node): Parents = [(3)], Siblings = [ ], rank =3 (fake rank)

Node 6: Parents = [(7)], Siblings = [8], rank = 19

Node 5: Parents = [(7)], Siblings = [8], rank = 19

(Note: In parent list the one with the parenthesis is the preferred parent.)

Packet is transmitted by node 6 is received by node 7 since the node 7 is malicious it drops the packet. So the Throughput in this scenario is 0.