

# Intrusion detection system in MANET

**Software:** NetSim Standard v14.4, Visual Studio 2022

## Project Download Link:

<https://github.com/NetSim-TETCOS/IDS-in-MANET-v14.4/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>

## Introduction

An Intrusion Detection System (IDS) functions as a security tool, diligently monitoring network activities for any unusual behavior and detecting suspicious activities. In our network scenario, when we intentionally introduce a malicious node, it disrupts the normal functioning of the network. The IDS identifies this irregularity and takes preventive action by blocking packet transmission to the malicious node. As a result, the packet transmission resumes its normal course, ensuring a secure and controlled network environment.

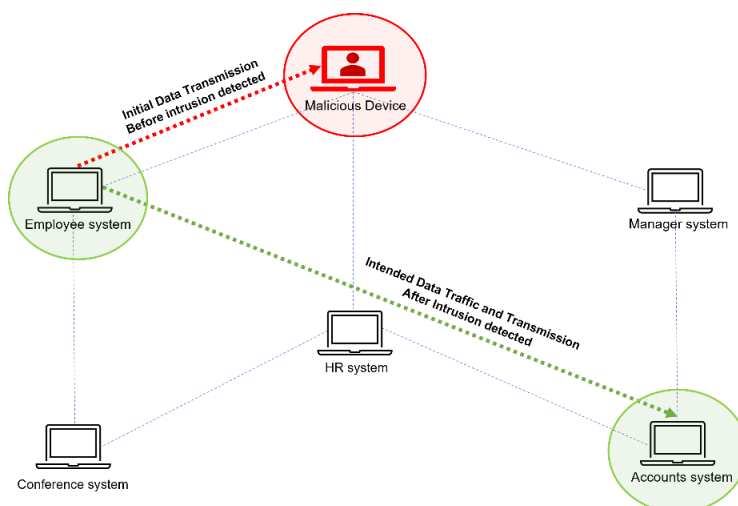


Figure 1: Intrusion detection system in a MANET office network

## Example

- The **IDS-MANETs\_v14\_4** workspace includes a sample network configuration that is already saved. To open this example, go to "Your Work" on the home screen of NetSim, and click on "IDS\_Experiment" from the list of experiments.

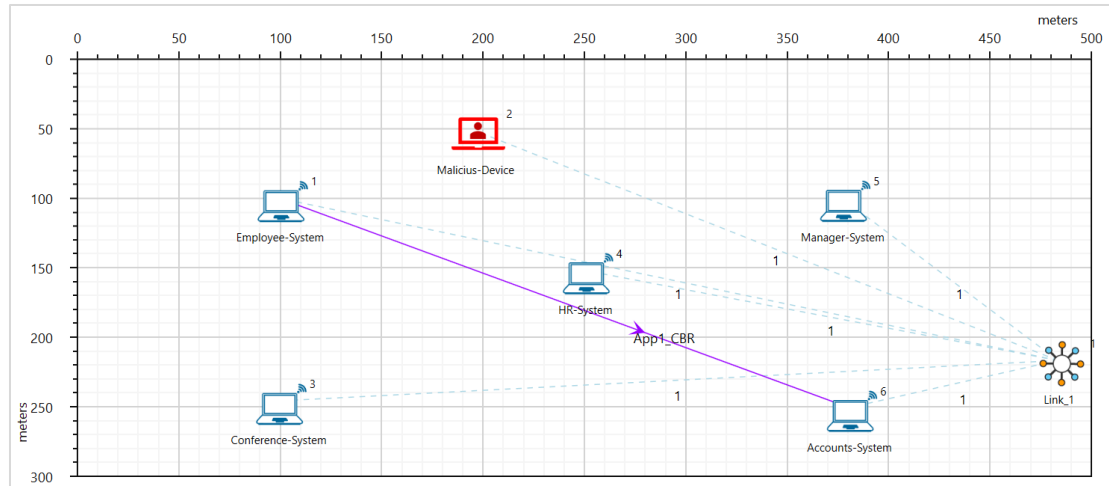


Figure 2: Network setup of Intrusion detection system in MANET

1. Wireless Link Properties
  - **Channel Characteristics:** Pathloss
  - **Path loss model:** Log distance
  - **Path loss exponent:** 2.5
2. An application is Created between Employee-system to Account-system and other properties are default.
3. Run the simulation for 100 seconds.

## Results and discussion

The time at which a malicious node is detected can be obtained from the custom metrics (IDS metrics). These metrics can be accessed by navigating to the additional metrics section, where you can find the start time (the time from which a node becomes malicious) and the detection time (the time at which the node was added to the blacklist).

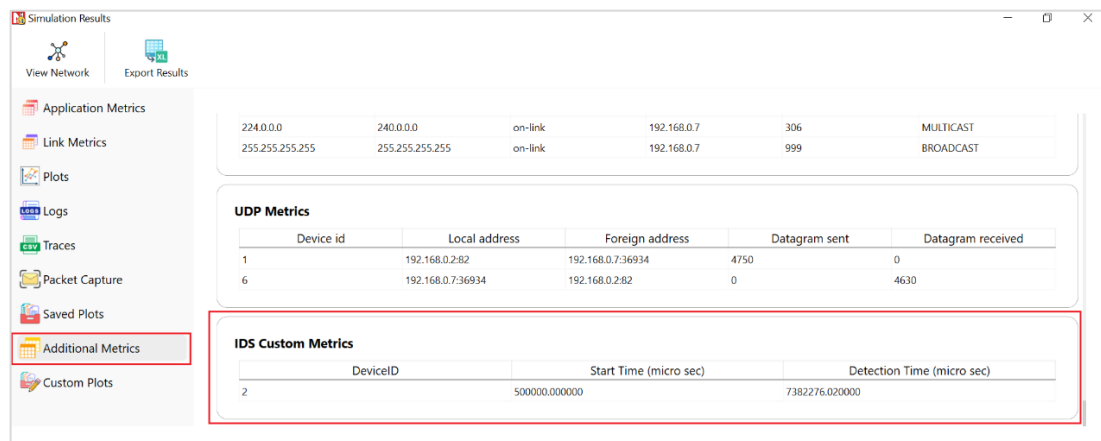


Figure 3: Dedicated Metrics for IDS

## Analysing results with Throughput vs Time Plot

After the simulation, click on the 'Throughput vs Time' plot in the results window. Initially, you will notice that all traffic flows to the malicious nodes, causing the throughput to remain at zero up until 7.38 seconds. Once the malicious node is detected and the route is changed, the throughput increases.

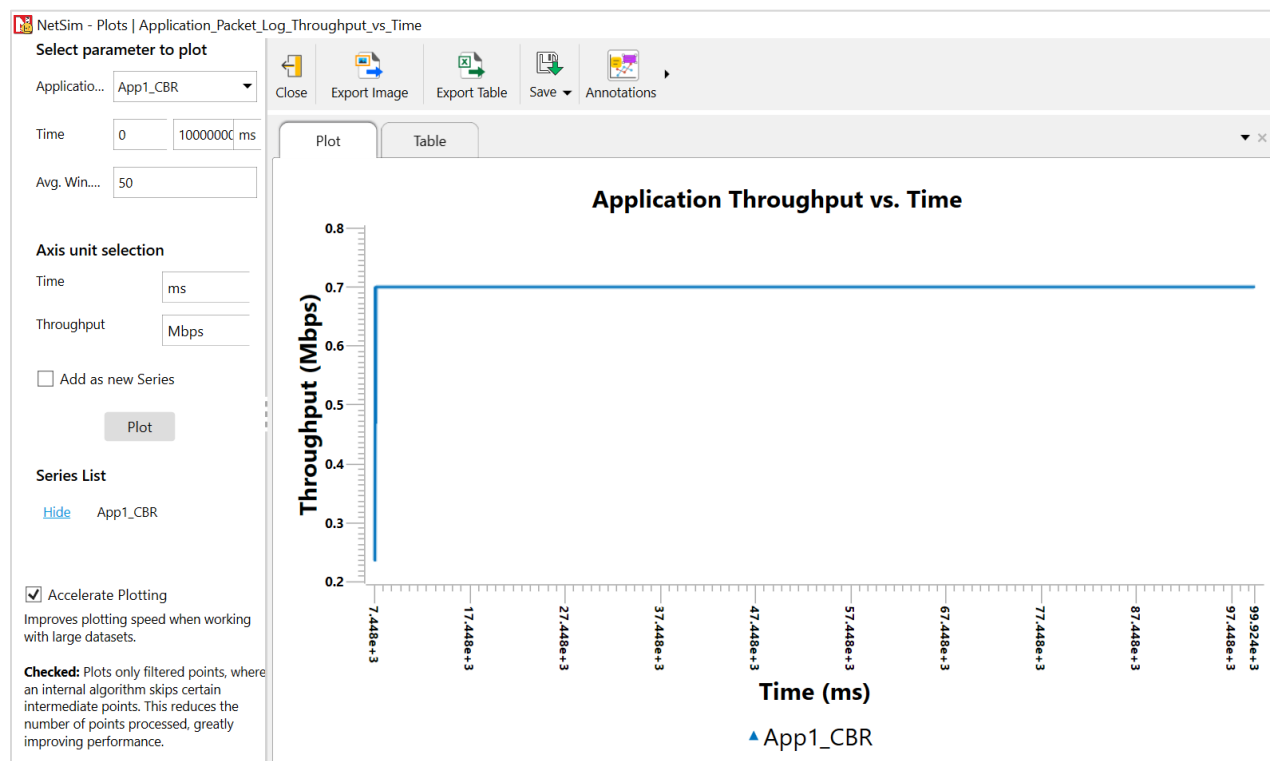


Figure 4: Initially, all traffic flows to the malicious node, resulting in a zero throughput until approximately 7.38 seconds. Upon detection of the malicious activity and subsequent rerouting of traffic, the throughput shows a significant increase, indicating the normal data flow.

## Analysing results with Packet trace

In the packet trace, we initially notice that all traffic flows to the malicious nodes. According to the original code setting, the watchdog timer is set to 2 seconds, and the failure threshold is set to 20 packets. Therefore, you would notice that around 7.38 seconds, the malicious node is detected, and the route to the destination changes in the subsequent route discovery process.

119	0 CBR	App1_CBR	NODE-1	NODE-6	NODE-1	NODE-2	73600
120	0 CBR	App1_CBR	NODE-1	NODE-6	NODE-1	NODE-2	73800
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-2	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-3	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-5	N/A
0 N/A	Control_Packet	DSR_RREP	NODE-2	NODE-1	NODE-2	NODE-1	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-2	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-3	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-5	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-1	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-2	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-3	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-5	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-6	N/A
0 N/A	Control_Packet	DSR_RREP	NODE-2	NODE-1	NODE-2	NODE-1	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-5	NODE-1	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-5	NODE-2	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-5	NODE-4	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-5	NODE-6	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-3	NODE-1	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-3	NODE-2	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-3	NODE-4	N/A
0 N/A	Control_Packet	DSR_RREQ	NODE-1	Broadcast-0	NODE-3	NODE-6	N/A
0 N/A	Control_Packet	DSR_RREP	NODE-6	NODE-1	NODE-6	NODE-4	N/A
0 N/A	Control_Packet	DSR_RREP	NODE-6	NODE-1	NODE-4	NODE-1	N/A
121	0 CBR	App1_CBR	NODE-1	NODE-6	NODE-1	NODE-4	74000
121	0 CBR	App1_CBR	NODE-1	NODE-6	NODE-4	NODE-6	74000

- At approximately 7.38 seconds, the intrusion detection mechanism identifies the presence of a malicious node, prompting the initiation of the route discovery process.
- Subsequently, despite attempts by the malicious node (Node-2) to send DSR Route Reply packets to the source, these actions are disregarded by the system due to the prior detection by the Intrusion Detection System (IDS).
- Importantly, DSR Route Reply packets from the real destination are considered after 7.38 seconds. Following the IDS detection, data transmission then resumes normal operations to the intended destination.

The following steps show how a user can run the IDS in NetSim to detect a malicious node and then set up a new route to the destination avoiding the malicious node:

- ## Malicious.c

```
fn_NetSim_DSR_MaliciousNode(); // This function is used to identify whether a current device is
malicious 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 received packets if the device is malicious, instead of forwarding the packet to the next hop.

## Pathrater.c

This file contains code for avoiding the malicious node and finding a new route (once the IDS detects the malicious node) in networks running DSR in Layer 3. Note that this system works only for UDP and not for TCP, since TCP involves receiving acknowledgments from the destination.

If `_NETSIM_PATHRATER_` is defined, the code is used to validate routes. When the node is identified as a malicious node and a route reply is processed, the function verifies the route reply in the route cache and checks for the blacklisted.

i.e., malicious node. When a malicious node is found that route entry is deleted from the cache.

## Watchdog.c

This file contains code for the IDS (Intrusion Detection System) and is added to the IEEE802\_11 project operating in Layer 2.

If `_NETSIM_WATCHDOG_` is defined, a watchdog timer starts the moment a packet is sent. Once the packet is forwarded to the next hop node, the current node checks the watchdog timer duration to determine whether the packet is being forwarded further to the destination node.

The malicious node does not forward packets that it receives. When the watchdog timer in the node (which forwarded the packet to the malicious node) expires, a counter measures the number of times the watchdog timer expires (in other words, the number of packets sent out but not forwarded by the next hop node). Once this counter's value reaches the failure threshold, the next hop is marked by the current node as a malicious node.

## References

- [1] Khaled Mohammed Saifuddin et al., "Watchdog and Pathrater based intrusion Detection System for MANET," 4<sup>th</sup> International Conference on Electrical Engineering and Information & communication Technology, 2018 IEEE.