Intrusion detection system in MANET

Software: NetSim Standard v14.2, Visual Studio 2022

Project Download Link:

https://github.com/NetSim-TETCOS/IDS_MANETs-v14.2/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.

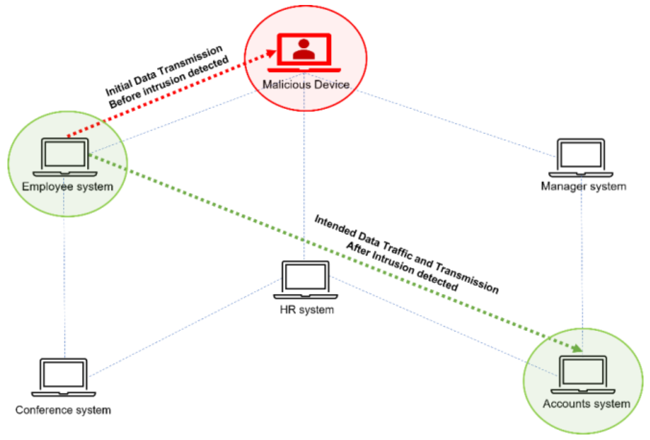


Figure 1: Intrusion detection system in a MANET office network

Example

- The IDS-MANETs-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.

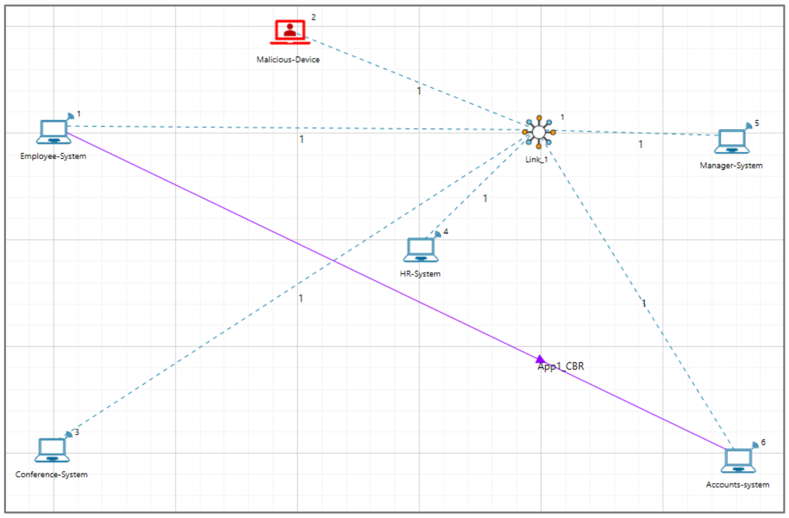
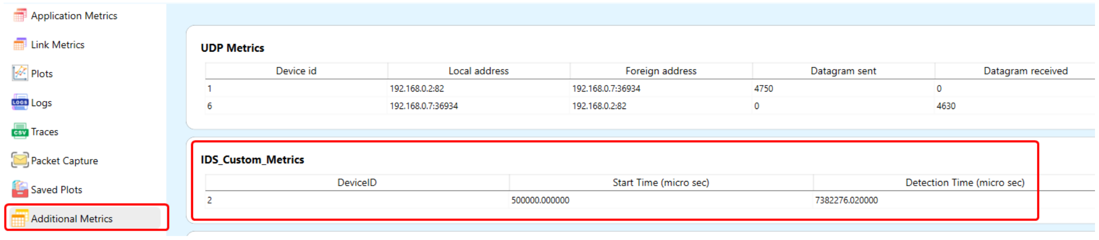


Figure 2: Network setup of Intrusion detection system in MANET

1. Wireless Link Properties
   - **Channel Characteristics :** Pathloss only
   - **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).



UDP Metrics

Device id	Local address	Foreign address	Datagram sent	Datagram received
1	192.168.0.2:82	192.168.0.7:36934	4750	0
6	192.168.0.7:36934	192.168.0.2:82	0	4630

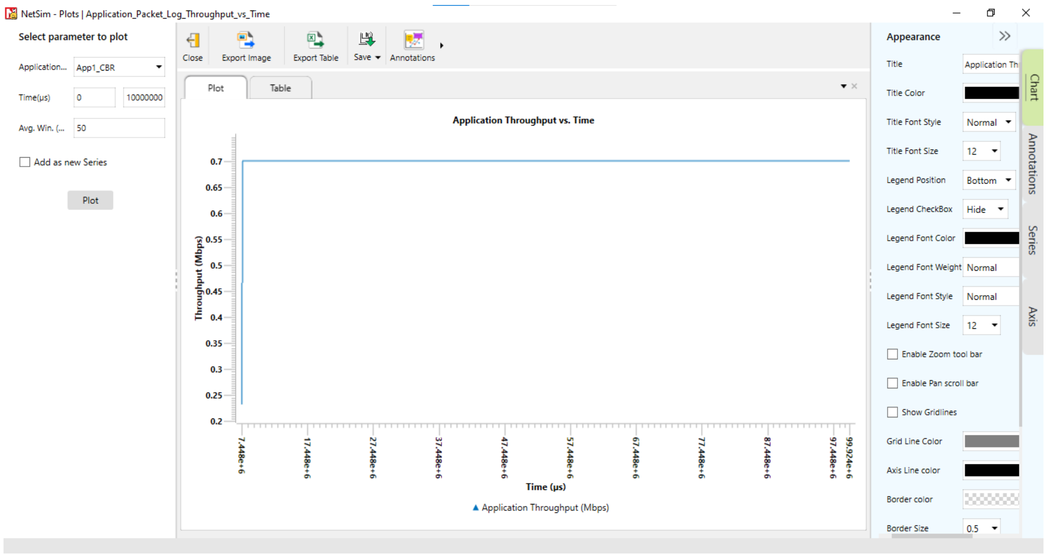
IDS_Custom_Metrics

DeviceID	Start Time (micro sec)	Detection Time (micro sec)
2	500000.000000	7382276.020000

Figure 3: Dedicated Metrics for IDS

Analyzing 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.


NetSim - Plots | Application_Packet_Log_Throughput_vs_Time
Close
Export Image
Export Table
Save
Annotations
Plot
Table
Plot
Application Throughput vs. Time
Throughput (Mbps)
Time (µs)
Application Throughput (Mbps)
Chart
Annotations
Series
Axis

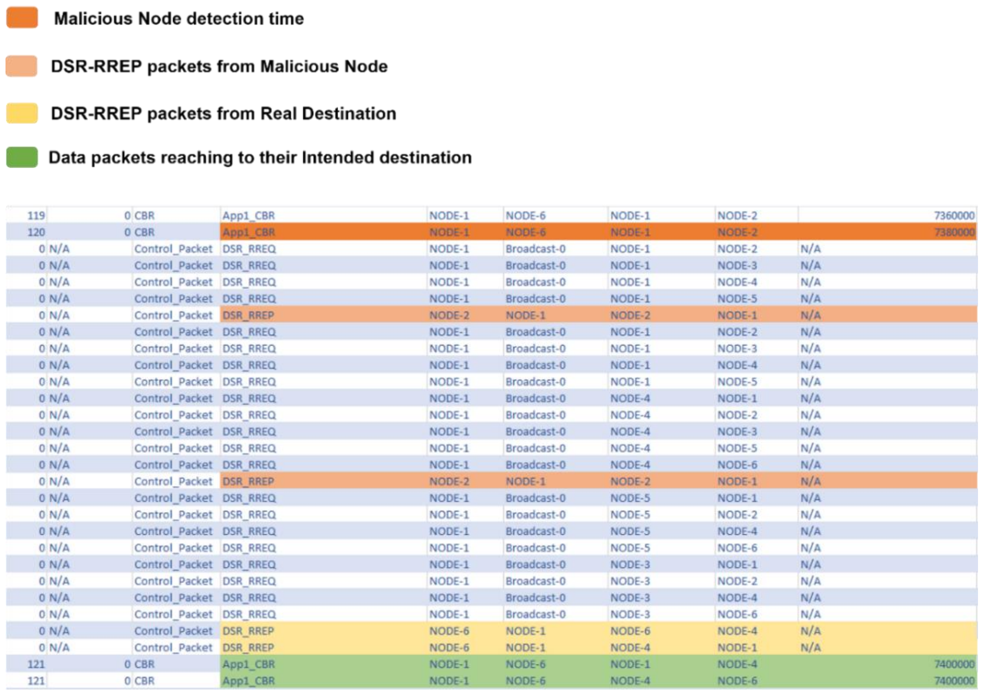
Select parameter to plot
Application... App1_CBR
Time(µs) 0 10000000
Avg. Win. (... 50
Add as new Series

Appearance
Title: Application Th
Title Color
Title Font Style: Normal
Title Font Size: 12
Legend Position: Bottom
Legend CheckBox: Hide
Legend Font Color
Legend Font Weight: Normal
Legend Font Style: Normal
Legend Font Size: 12
Enable Zoom tool bar
Enable Pan scroll bar
Show Gridlines
Grid Line Color
Axis Line color
Border color
Border Size: 0.5

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.

Analyzing 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.



🟧	Malicious Node detection time
🟧	DSR-RREP packets from Malicious Node
🟨	DSR-RREP packets from Real Destination
🟩	Data packets reaching to their Intended destination

119	0	CBR	App1_CBR	NODE-1	NODE-6	NODE-1	NODE-2		7360000
120	0	CBR	App1_CBR	NODE-1	NODE-6	NODE-1	NODE-2		7380000
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		7400000
121	0	CBR	App1_CBR	NODE-1	NODE-6	NODE-4	NODE-6		7400000

Figure 5: Analysis of results using packet trace

- 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.

Files Used in this project

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:

- Creating malicious nodes for a particular network scenario is explained in the Malicious.c file.
- To detect the intruder and send data via a new route, the following files are added: Pathrater.c and Malicious.c in DSR, and Watchdog.c in IEEE802_11.

Malicious.c

This file contains code for turning a node into a malicious node:

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," 4th International Conference on Electrical Engineering and Information & communication Technology, 2018 IEEE.