Intrusion detection system in MANET

Software: NetSim Standard v14.1, Visual Studio 2022

Project Download Link:

https://github.com/NetSim-TETCOS/Intrusion-Detection-System-in-MANET-v14.1/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 upon 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 transmission of packets resumes its normal course, ensuring a secure and controlled network environment**.**

Real-World Context:

Consider a scenario within a standard office network. An employee intends to share sensitive documents with the Accounts system. However, the presence of a Malicious Device in the network leads to an initial misdirection of data towards the malicious entity. The Intrusion Detection System (IDS) plays a crucial role in this context. Upon detecting the malicious behavior, the IDS intervenes to rectify the issue, redirecting the data to its intended destination – the Accounts system. In this narrative, the IDS acts as a vigilant security measure, ensuring the secure and accurate transmission of sensitive information, even in the presence of potential threats

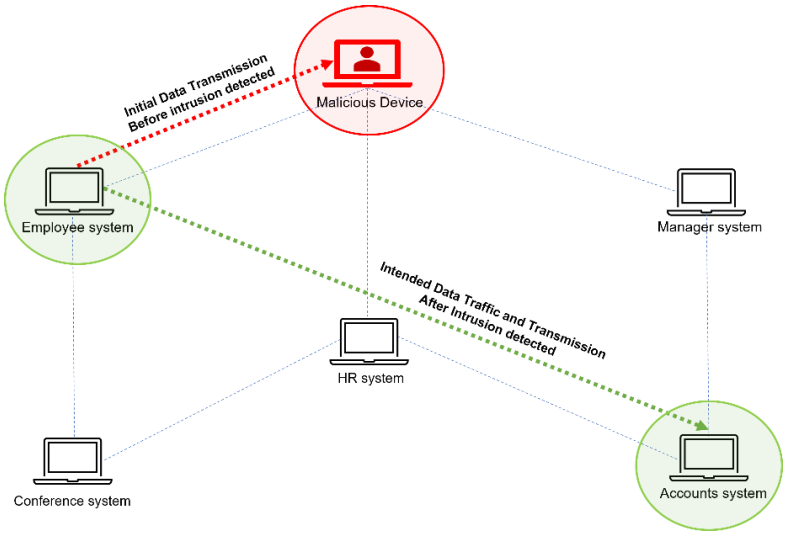

Figure 1: Real world scenario for Intrusion Detection System

Intrusion Detection System Overview

- The IDS continuously monitors network activity for abnormal patterns or malicious behavior.
- When a potential intrusion is detected, the IDS initiates an immediate response to safeguard the network.
- For instance, Here Employee System wanted to send some sensitive information to the Accounts system, the IDS analyzes that the data is being transmitted to malicious Device.
- The IDS dynamically responds to detected threats by redirecting the data flow back to its intended destination and preventing unauthorized access.
- By identifying and mitigating potential risks, the IDS enhances the overall resilience and security of the network.
- Overall, the IDS makes the network stronger and safer by finding and fixing potential issues right away.

NetSim's Role:

We utilize NetSim to simulate and analyze security scenarios, focusing on the Intrusion Detection System (IDS). NetSim provides a controlled virtual environment for our study, allowing us to replicate real-world situations and evaluate the performance of the IDS in detecting potential threats. This document will detail our project's specific objectives related to the integration and analysis of the IDS within NetSim.

Example:

- The IDS-MANETs-WorkSpace comes with 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 the **IDS_Experiment** from the list of experiments.

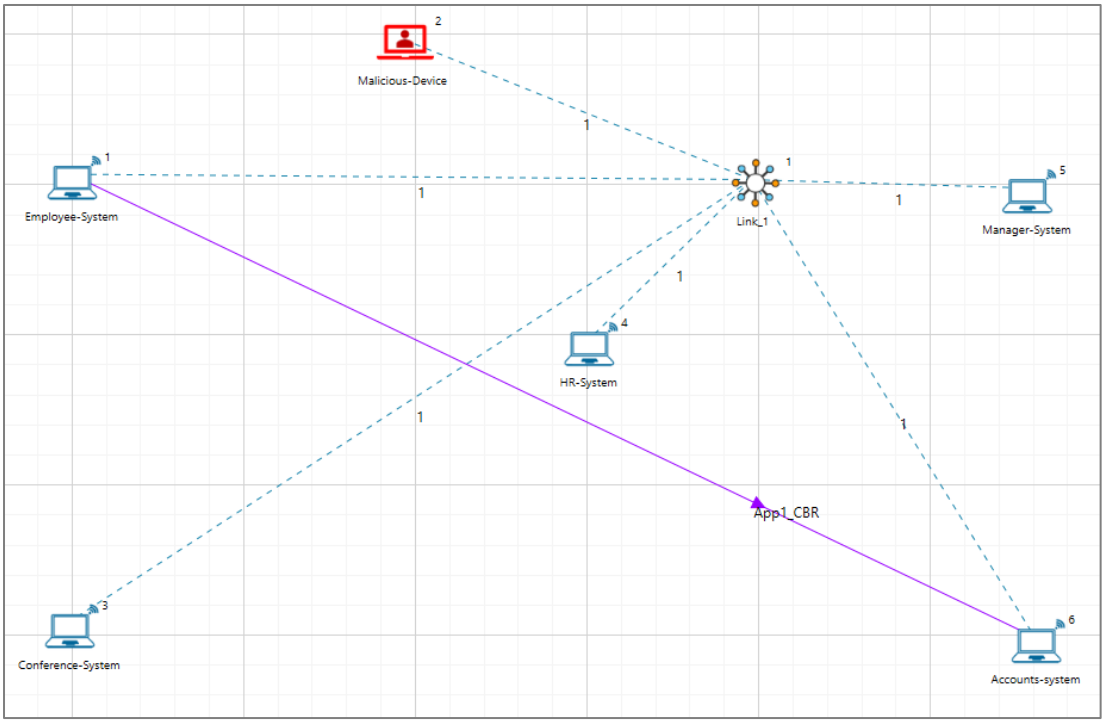

Figure 2: Network setup of Intrusion detection system in MANET

1. Wireless Link Properties
   - Channel Characteristics - Pathloss only
   - Path loss model - LOG_DISTANCE
   - Path loss exponent – 2.5
2. An application is Created between Wireless_Node_1 to Wireless_Node_6 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).

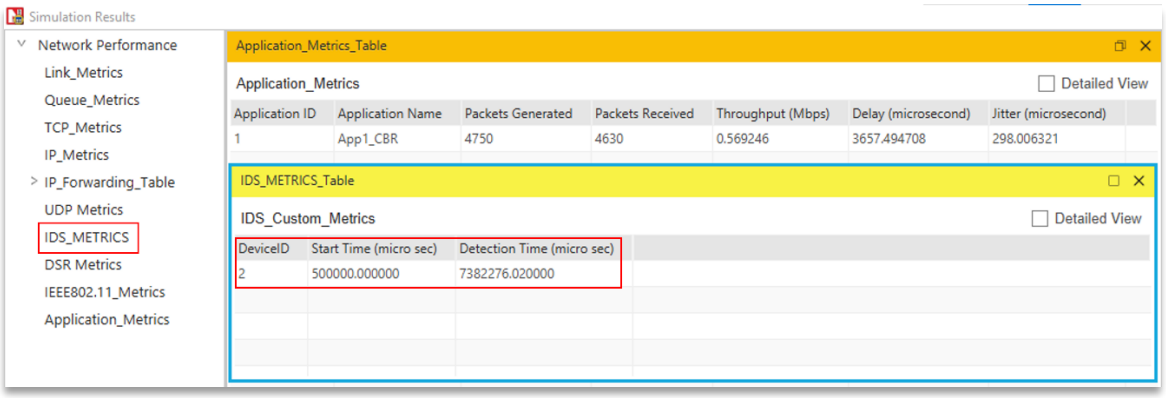

Figure 3: Dedicated Metrics for IDS

Analyzing results with Packet trace

In the packet trace, we can notice initially all traffic would flow to the malicious nodes. As Per the original code setting the Watchdog timer is set to 2 seconds and the failure threshold is set to 20 packets. So, you would notice that around 7.38 seconds, the malicious node is detected and the route to destination would change 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 4: 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, given the prior detection by the Intrusion Detection System (IDS).
- Importantly, **DSR Route Reply** packets from the real destination are considered after the 7.38-second. Following the IDS detection, the data transmission then resumes its normal operation 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 setup a new route to the destination avoiding the malicious node.

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

Malicious.c

This file contains code for turning a node into malicious node

- **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.

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 would work only for UDP and not for TCP, since TCP involves receiving ack's from the destination.
If _NETSIM_PATHRATER_ is defined, the code is used to validate routes. When the
Node is a Malicious Node, and a Route Reply is processed, the Function verifies the route reply in the route cache and checks for the blacklisted node.
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 in IEEE802_11 project operating in Layer 2.

If _NETSIM_WATCHDOG_ is defined, a watchdog timer starts the moment a packet is sent. Once a packet is forwarded to next hop node, the current node checks for watchdog timer duration if the packet is getting forwarded further on to destination node or not.

The malicious node does not forward packets that it receives. The watchdog timer in the node (which forwarded the packet to the malicious node) expires. A counter is present which 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 hope is marked by the current node as a malicious node.