DIS Flooding Attack in IOT Networks Running RPL

Software Recommended: NetSim Standard v13.0 (64 bit), Visual Studio 2019

Reference: Y. Mai, F. M. Rodriguez and N. Wang, "CC-ADOV: An effective multiple paths congestion control AODV," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2018, pp. 1000-1004.

Secure URL for the GitHub repository: https://github.com/NetSim-TETCOS/RPL_DIS_Flooding_v13_0/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-upnetsim-file-exchange-projects

Introduction:

In RPL, DIS messages are used by nodes to join the network. A node sends a DIS message to its neighbour nodes to request the routing information so that it may join the existing DODAG. Thus, a new node continuously transmits DIS messages with a fixed interval until it receives a DIO message from any neighbour node. Once a node receives a DIO message, it stops transmitting DIS messages and joins the network by sending DAO to the solicited node.

A malicious node can utilize this feature to degrade the network performance by choosing different DIS transmission interval for periodically transmitting DIS messages to its neighbouring nodes; this is called a DIS flooding attack. This leads to an increase in the network's control packet overhead and power consumption.

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 instead of joining existing DODAG it just Drops DIO and frequently transmits DIS messages. Which forces normal nodes to reset their trickle timers and flood the network with DIO messages.

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.

2. rpl_drop_msg()

This function is used to drop the DIO messages received by the malicious nodes instead of replying with a DAO message.

You can set any sensor 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_MaliciosNode() function.

Steps:

1. Go to home page, Click on Your work \rightarrow Workspace options \rightarrow Open code

NetSim Home									- • ×
NetSim Stanc Network Simulation/Emulat Version 13.0.22 (64 Bit)	lard ion Platform								N
New Simulation	Ctrl+N	Current workspace: Workspace_Dis	FLOOD_RPL	Matumak turus	Size		Q	Experiment name	
Your work	Ctrl+O	DIS ELOOD Case? Example	24-02-2021	internet of Thions	47 KR	View Results	Ferrent	*	
		DIS FLOOD Case1 Example	23-02-2021	Internet of Things	45 KB	View Results	Export	а а	
Examples	Ctrl+E								
License Settings Exit	Alt+F4								
Support Ansert/FAQ ContexTechnical Support Fmal - support Stations con		Open Cude Reve	ef Code Reset Bravies Learn Videos Experiment Manual	Mare Options	Docurrentation User Manual Technology Liberies Source Code Help		Contact Us Intel - saled@eteos.com Phone - +91 757 605 4221	Rea Activate Windows Go to Settings to activate Win	ndows.

- a. Expand RPL project and open Malicious.c file.
- b. Set malicious node id.



2. Add the code that is highlighted in RPL_Message.c file

Sen	DIS.c* 7	RPL_Message.c 🗢 🗙 Neighbor.c	Malicious.c*	RPL.c	DODAG.c	DIO.c	Trickle.c	RPL.h		
/er E	💁 RPL		-	(Global Scope)				 	sg()	
- pla	425	pevent.nDeviceType = DEVIC	, CE TYPE(ndevid);							
ē	426	pevent.nEventType = NETWOR	RK_OUT_EVENT;							
=	427	<pre>pevent.nPacketId = packet</pre>	->nPacketId;							
ě	428	pevent.nProtocolId = NW_PF	ROTOCOL_IPV6;							
ğ	429	<pre>pevent.pPacket = packet;</pre>								
	430	<pre>fnpAddEvent(&pevent);</pre>								
	431	}								
	432									
	433	<pre></pre>								
	434	{								
	435	switch (pstruEventDetails)	->pPacket->nCont	trolDataType :	6 100)					
	436				-					
	437	<pre>case DODAG_Information_Object:</pre>								
	438	if (tn_NetSim_RPL_MaliciousNode(pstruEventDetails))								
	459	rp1_drop_msg();								
	440	else								
	441	<pre>process_did_msg();</pre>								
	443	case Destination Advertise	ement Object:							
	444	rpl process dao msg()	:							
	445	break;	,							
	446	case DODAG Information So	licitation:							
	447	rol process dis mse():								
	448	break:								
	449	default:								
	450	fnNetSimError("Unknown	n rpl ctrl msg 9	6d in %s",						
	451	<pre>pstruEventDetails->pPacket->nControlDataType,</pre>								
	452	FUNCT	TON 1.							
	100 % 🔹	₩ 0 <u>∧</u> 2 ¢ → <								

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

If if to all all all all all all all all all al	jie Edit Yew Project Build D	nug Tegt Agelyze Jools Egtensions Window Help Search (Ctri-Q	NetSim				0	- o
Name of the loss of the	- © 🗄 - 🖕 🖬 📲 🏷 - 🤇 - 🗍	ebug 🔹 x64 🔹 🕨 Local Windows Debugger - 🛛 🏓 🖾 🖉	@[고왕]# 영영영 -					남 Live Share
Pre- Construction 0 Construction 0 Construction Construction Construction Construction Construction Construction Con	L_Missage.c = × RPL.c DAO.c	Maticious: FPLh				Solut	ion Explorer	Ţ
41 memer(lacent, s, histor permi); 42 permin(lacent, s, histor permi); 43 permin(lacent, s, histor permi); 44 permin(lacent, s, histor permi); 45 permin(lacent, s, histor permi); 46 permin(lacent, s, histor permi); 47 permin(lacent, s, histor permi); 48 permin(lacent, s, histor permi); 49 permin(lacent, s, histor permi); 40 permin(lacent, s, histor permi); 41 permin(lacent, s, histor permi); 42 permin(lacent, s, histor permi); 43 permin(lacent, s, histor permi); 44 permin(lacent, s, histor permi); 45 permin(lacent, s, histor permi); 46 permin(lacent, s, histor permi); 47 permin(lacent, s, histor permi); 48 permin(lacent, s, histor permi); <	191	 (Global Scope) 		 Ø spl_nede_send_msg(NETSIM_ID nde 	ovid, NetSim_PACKET* packet)	. 01	୦ 🙆 📅 • 🏷 ୬ - ଅକ୍ର 🖉 🔯	P -
121 promitboxit.iterproduction: providproduction: providpr	421 menset(&pes	ent, 0, sizeof pevent);				+ Seerc	h Solution Explorer (Ctri+;)	
44 permit ubertstate = made() 45 permit ubertstate = made() 46 permit ubertstate = made() 47 permit ubertstate = made() 48 permit ubertstate = m	422 pevent.dEve	itTime = pstruEventDetails->dEventTime;				- F	Solution 'NetSim' (2 of 2 projects)	
<pre>44 44 45 46 46 47 47 47 47 47 47 47 47 47 47</pre>	423 pevent.dPac	<pre>cetSize = fnGetPacketSize(packet);</pre>				P I	5 IP	
<pre>45 provent.abscirts.prove EUVIC_INF (unit); 45 provent.abscirts.prove EUVIC_INF (unit); 46 provent.abscirts.prove EUVIC_INF (unit); 47 provent.abscirts.prove EUVIC_INF (unit); 48 provent.abscirts.prove EUVIC_INF (unit); 49 provent.abscirts.prove EUVIC_INF (unit); 49 provent.abscirts.prove EUVIC_INF (unit); 40 provent.absci</pre>	424 pevent.nDev	<pre>iceId = ndevid;</pre>				. 17	* 000 * 0-20	
<pre>44 45 46 45 45 45 45 45 45 45 45 45 45 45 45 45</pre>	425 pevent .nDev	<pre>iceType = DEVICE_TYPE(ndevid);</pre>				1.1	Rebuild	
100 potent understall a = ak unders	420 pevent.nEve	ICTYPE = NETWORK_DUT_EVENT;				4.0	Care	
<pre>44 provide rule and rule rule rule rule rule rule rule rule</pre>	427 pevent.nPac	<pre>letid = packet->nPacketid;</pre>					Man	
do do do do do do do do do do	420 pevent.nPro	IDEDIIG = NW_PROTOCOL_IPV0;					tester and Cade Classes	
adia mplaneterin (Weevent); adia provid rel.process_ctrl.msg() adia f adia provid rel.process_ctrl.msg() adia f adia provid rel.process_ctrl.msg() adia provid rel.process_ctrl.msg(); adia provid rel.msg();	429 pevent.pPac	<pre>tet = packet;</pre>					Deviced Code Creation	
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	430 ThpAddevent	(apevent);					Property Carry Reterrat Residents	
141 genetal rel process_ctrl_msg() 429 [sulth (pstructurellballs-speaket-srcantrollbals/pr % 108) 439 [crc DOML Information Digits: 439 [crc DomL Information Digits: 4	431 []						Description	
<pre>34</pre>	432						Scope to This	
<pre>int int () () () () () () () () () () () () ()</pre>	455 Evolu rp1_proces	2_ccu1_msg()				6	B: New Solution Explorer View	
20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ANA IL suiteb (mit	enformtDataile seBecket sefertes[DataTure # 100)					Build Dependencies	
tor the second	435 E Switch (pst	ueventuetalls-speacket-shcontrolloatalype w 100)				50	Add	
1 (f (f) = Single	430	Tefermation Objects					Class Wigard	Cel+SH
1 1 (Un splange.gg); 1 1 (U	432	NotSin DDL MaliciousNode(nstruEventDatails))				100	Manage NuGet Packages	
<pre>compared compared compare</pre>	430	rel deen mrs():				i i i	S Gat as Start Lin Deniant	
and a process die med (); 433 (sone Destination, Aberetiseen Coligie (); 434 (sone Destination, Aberetiseen Coligie (); 435 (sone Destination, Aberetiseen Coligie (); 436 (sone Destination, Aberetiseen Coligie (); 437 (sone Destination, Aberetiseen Coligie (); 438 (sone Destination, Aberetiseen Coligie (); 439 (sone Destination); 439 (sone Destination); 430 (sone Destination); 440 (s	439	rpi_arop_msg();				8.	er belæs ogstop engett	
tag car be day	440 8158	manage dia mar():						
And Constraint of the second of the	443	_process_uio_msg();					5 Cut	Cel+X
444 Lorenzy (Lorenzy Generges)); A failing (Lorenzy Generges); Participation 453 Lorenzy (Lorenzy Generges); Failing (Lorenzy Generges); Participation 464 Lorenzy (Lorenzy Generges); Failing (Lorenzy Generges); Participation 465 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 466 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 467 Construction Participation Construction 468 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 469 Construction Participation Construction 469 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 469 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 469 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 460 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 461 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 462 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction 463 Lorenzy (Lorenzy Generges); Lorenzy (Lorenzy Generges); Construction </td <td>442 Dreak,</td> <td>ation Admeticement Objects</td> <td></td> <td></td> <td></td> <td></td> <td>Baste</td> <td>Cb1+V</td>	442 Dreak,	ation Admeticement Objects					Baste	Cb1+V
and breaks the set of	444 case besch	core dao men():					K Remoye	Del
dia dia dia dia dia dia dia dia	445 break:	.ess_ueo_msg(),				- 1	Rename	
tor p p p (p = (p = (p = (p = (p = (p = (p	445 Dreak;	Tefernahien Colisitations					Unjoad Project	
tati i formati	447 Case DODAG	cose down in ormacian solicitation.					Load Project Dependencies	
Code Becyclon Code Be	449 hpeaks	.ess_uis_usg();				- 50	Rescan Solution	
tati - 1 a Opper Propertie - Others A 1x12 Wannys O 2Manups 1 bed - relations - Unit Pro- * Cate Ideoption - Properties - Unit Pro- * Cate Ideoption - Properties - Unit Pro- * Cate Ideoption - Properties - Unit Pro- * Cate Ideoption - Pro-	5 · 00 42 6 -				Enc.425 Cite.41 Crit.46 Table Cit.	1	Display Browsing Database Em	05
Catholing & Maria Thomas 0 Billionging (%) Devision - South horizon P. 2 Southet Horizon P. 2 Southet Hor	at at		- 1	C Dated			Clear Browsing Database Error	4
* Cate Boundon // Propert He Let ↓ Visual Statis XXII system Visual	tin Solution - O Church	0 of 2 Warrings 0 0 Messages 😽 Build a Intelligence 🔹	Search Finan List R	Show output from:			Open Folder in File Explorer	
Visual Randow 500 mpalane Visual Malific Appropriate Appropriat	" Code Description	Project	File Line				P Properties	Alt+ Ent
Go to Settings to activate Windows					*	Visual St Version 1	uelia 2019 update 15.86 is downloadicel and ready to	install,
							Go to Settings to activate V	finderweiter

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

Settings that were done to create the network scenario for DIS_Flooding Attack.

- 1. Create a network scenario in Internet of Things with the Transport Protocol as UDP.
- 2. For example, you can create a scenario as shown in the following screenshot:



Source – Device id 2 Destination – Device id 8 Sinkhole (malicious node) – Device id 1

Link Properties (Adhoc link1)

Channel characteristics – Path Loss only Path Loss model – LOG DISTANCE Path Loss Exponent: 3.5

Device Properties: Go to Sensor Properties -> Network_Layer -> DIS_Interval -> 10ms

- **3.** Run the Simulation for 100 seconds.
- **4.** View the packet animation. You will find that the malicious node (Device id 1) even after receiving DIO from neighbour nodes instead of joining existing DODAG it just Drops DIO and frequently transmits DIS messages.
- **5.** This will have a direct impact on the Application Throughput and Delay which can be observed in the Application Metrics table present in NetSim Simulation Results window.

Case 1: Application throughput Vs. Application generation rate

We fix the DIS interval to 10 milli seconds and vary the application generation rate to see the impact of DIS flooding on the network performance.

	Through	out (Mbps)	Delay (ms)		
Generation Rate (Kbps)	With_DIS	Without_DIS	With_DIS	Without_DIS	
60	0.06	0.06	1880.97	51.80	
80	0.06	0.08	9899.92	51.75	
100	0.07	0.10	15577.11	51.76	
120	0.07	0.12	19645.16	239.45	

This can be further understood with the help of following plots:





We can observe that the application throughput decreases in case of DIS flooding when compared with the usual simulations for various application traffic generation rates.

Delay is comparatively high in case of DIS flooding and increases with the increase in generation rate. This is because the nodes are busy receiving and responding to DIS messages from malicious node frequently. The nodes that receive DIS messages are forced to reset their trickle timers and flood the network with DIO messages.

Case 2: Application throughput Vs. DIS Interval Time

We fix the application generation rate to 250 Kbps and vary the DIS interval to see the impact of DIS flooding on the network performance.

DIS Interval (ms)	Throughput (Mbps)
25	0.10
20	0.10
15	0.09
10	0.08
5	0.08

This can be further understood with the help of following plots:



We can observe that the application throughput decreases as we decrease DIS Interval time. Upon decreasing the DIS interval, more DIS messages will be sent by the malicious nodes more frequently. Legitimate sensors spend more time in processing and responding to DIS messages than sending the data packets.

DIS flooding severely degrades the performance of Low Power and Lossy Networks (LLNs) because of the increase in control packet overhead.