Addressing the DAO Insider Attack in RPL's Internet of Things Networks

Baraq Ghaleb, *Student Member, IEEE,* Ahmed Al-Dubai, *Senior Member, IEEE,* Elias Ekonomou, Mamoun Qasem, *Student Member, IEEE,* Imed Romdhani, and Lewis Mackenzie, *Member, IEEE*

***Abstract*—In RPL routing protocol, the DAO (Destination Advertisement Object) control messages are announced by the child nodes to their parents to build downward routes. A malicious insider node can exploit this feature to send fake DAOs to its parents periodically, triggering those parents, in turn, to forward the fake messages upward to the root node. In this study, we show how this behavior can have a detrimental side effect on the performance of the network, increasing power consumption, latency and reducing reliability. To address this problem, a new scheme is introduced to mitigate significantly the effect of the DAO attack on network performance.**

***Index Terms*—Internet of Things, Low-power and Lossy Networks, RPL Security, DAO Attack.**

I. Introduction

Recently the Low-power and Lossy Networks (LLNs), a collection of interconnected tiny sensor nodes, have been considered one of the key enabling blocks of the ever-growing Internet of Things paradigm [1] [2]. Due to their scarce resources, the Internet Engineering Task Force (IETF) has specified the IPv6 Routing Protocol for LLN (RPL) [3] as the routing standard for such networks [3][4][5]. Since it was a proposal, the RPL's security aspects have been analyzed by several research efforts reporting the existence of multiple security concerns that need to be addressed in order to facilitate the adoption of the protocol in a wide range of applications [6][7][8][9][10][11][12][13][14]. One of such security concerns is the DAO (Destination Advertisement Object) attack, where a compromised node sends periodic DAO messages to its parent nodes forcing them, in turn, to flood the network with DAO messages, an action that can severely harm energy efficiency, latency and reliability of the entire network. In fact, unlike other control-based attacks, DAO messages are transmitted in end-to-end fashion, from the sensor node toward the root (the details of the exact mechanisms are explained in Section II), so the level of damage is not restricted to the local scope of the attacker. Indeed, a DAO message sent by a child leaf node located on the edge of the network will trigger network-wide DAO transmissions because the DAO must be forwarded by every intermediate parent between that child and the network root affecting network performance and consuming its resources [3][15]. To address this issue, a new simple, yet effective solution has been proposed in this article with the goal to mitigate the effect of DAO insider attack on the performance of RPL's IoT networks. The acquired results carried out by means of simulation experiments have demonstrated the capacity of the proposed solution in mitigating the attack and almost restoring back the perceived efficiency of RPL in terms latency, overhead, energy consumption and packet delivery ratio.

The remainder of this paper is organized as follows. Section II gives a brief overview of the RPL protocol highlighting its routing mechanism to build downward routes. Section III introduces a description of the DAO attack, analyzing its effect on the network. The proposed mitigation mechanism is introduced in Section IV. The detail of the protocol evaluation and discussion is in Section V, while Section VI concludes the paper and discusses future work.

II. RPL Routing Protocol Overview

A. RPL Topology and Operations

RPL organizes its physical network into a form of Directed Acyclic Graphs (DAGs) where each DAG is rooted at a single destination and is referred to as a Destination-Oriented DAG (DODAG) in RPL's terms [3][4][5]. RPL uses the term upward routes to refer to routes that carry the traffic from normal nodes to the LBR whereas routes that carry the traffic from the DODAG root to other nodes are called the downward routes [3].

To facilitate the upward traffic pattern, a DODAG topology centered at the network root must be constructed. In such a topology, each non-root node willing to participate in upward communication must select one of its neighbors to act as that nodes default route (DODAG parent) towards the root [3]. The construction of the DODAG starts with the root multicasting control messages called DODAG Information Objects (DIOs) to its RPLs neighbors. The DIOs carry the necessary routing information and configuration parameters required to build the DODAG [3][4]. An RPL node receiving a multicast DIO message will: (1) add the sender address to its candidate parent set; (2) calculate its distance (rank) with respect to the DODAG root based on the rank of that candidate parent, routing information advertised; (3) setup its default route (preferred parent); and (4) update the received DIO with its own rank and multicast it to other neighboring nodes, enabling them, in turn, to perform the previous operations [3][4].

To enable bi-directional communication, downward routes also need to be constructed. This is achieved by deploying another type of ICMPv6 control messages, namely, the Destination Advertisement Object (DAO). An RPL node willing to announce itself as a reachable destination from the root point of view, unicasts a DAO to its preferred parent advertising its own destination prefix. The processing of the received DAO by the parent relies on the current mode of operation advertised in the DIO messages. To this end, RPL has specified two modes for creating and maintaining downward routes, namely, storing (table-driven) and non-storing (source routing) [3][4].

In the storing mode, when a parent receives a DAO from one of its children, it: (a) stores the announced destination prefix locally

Manuscript received March 30, 2018; revised August 30, 2018; accepted October 12, 2018. Date of publication Month xx, xxxx; date of current version October 17, 2018. The associate editor coordinating the review of this paper and approving it for publication was B. Shihada. (*Corresponding author: Ahmed Al-Dubai.*)

B. Ghaleb, A. Al-Dubai, E. Ekonomou, M. Qasem and I. Romdhani are with the School of Computing, Edinburgh Napier University, Edinburgh, EH10 5DT, UK,(e-mail:b.ghaleb@napier.ac.uk; a.al-dubai@napier.ac.uk; e.ekonomou@napier.ac.uk; m.qasem@napier.ac.uk ; i.romdhani@napier.ac.uk).

L. Mackenzie is with the School of Computing Science, Glasgow, G12 8QQ UK. (e-mail: Lewis.Mackenzie@glasgow.ac.uk).

in its routing table along with the DAO sender address, as the next hop to reach that destination; and (b) forwards the received DAO, in turn, to its own preferred parent to ensure the propagation of the advertised destination upward to the DODAG root [3][4]. This process is repeated by each intermediate node until the DAO is finally received by the DODAG root.

In the non-storing mode of RPL, the same procedure is followed but a parent receiving a DAO does not store any routing state. Instead, it simply forwards the message to its own preferred parent until it is finally received by the DODAG root. Once the DODAG root receives the transmitted DAO, it records the source route of the intended destination for later use by the data-plane [3][4].

III. The DAO Attack

RPL uses DAO messages to build downward routes enabling bi-directional communication. The specification of RPL does not stipulate when and how often DAOs are transmitted. Thus, different implementations may opt to use different mechanisms to achieve this process. For instance, the study in [15] has opted to transmit periodically DAOs whereas the Contiki RPL implementation [16] transmits the DAO based on the Trickle timers of DIOs. In Contiki RPL, a child node should unicast a DAO to its preferred parent on three occasions: 1) upon receiving a DIO from that parent; 2) upon changing its preferred parent; and 3) upon detecting some specific errors.

An interesting point in this context is that the transmission of a DAO message by a child node will trigger the transmission of multiple DAOs proportional to the number of intermediate parent nodes between that child and the DODAG root. An adversary can exploit this fact to harm the network by repeatedly and judiciously (to go undetected) transmitting DAOs to its parent node. A simple way to mount this attack is to replay an eavesdropped DAO from a legitimate node by an outsider triggering DAO forwarding upward by the nodes parents [14]. This kind of attack can be mitigated using security services provided by the underlying layers or RPL itself such as MAC-layer encryption and the cryptographic challenge-response handshake [14]. However, these mechanisms will not be sufficient to counter an attack where the attacker is an insider or compromised node [14].

IV. The Proposed Solution

In order to address a DAO insider attack in RPL, a new mechanism has been proposed, named SecRPL that restricts the number of forwarded DAOs by a parent. In fact, there are two options for how this restriction can be applied: the first is to restrict the entire number of forwarded DAOs regardless of the source node (i.e. the node who initiated the DAO); the second is to restrict the number of forwarded DAO per destination.Here we opt to use the second option, as the first option would result in blocking some DAOs coming from non-attacker nodes effecting negatively the quality of the downward paths. It may also result in DAOs of some nodes being blocked more than DAOs of some others. In particular, each parent node associates a counter with every child node in its sub-DODAG. When the number of forwarded DAOs for a child exceeds a pre-specified threshold, the parent discards any DAO message carrying the prefix of the respective child.

To ensure that no node will be blocked due to the time factor, the counter is reset between each two consecutive DIOs. Specifically, when the parent node sends out a DIO message, the counters for all of its children are reset.

Algorithm 1 : DAO Insider Attack Countermeasure

```
procedure INITIALIZATION
    set $DAO\_For\_MAX$
end procedure

procedure DIO TRANSMITTED
    for each $child$ in $Children\_list$ do
        $child\_DAO\_Counter = 0$
    end for
end procedure

procedure CHILD'S DAO RECEIVED
    if $child\_DAO\_Counter < DAO\_For\_MAX$ then
        forward the child DAO
        $child\_DAO\_Counter$++
    else
        discard the child DAO
    end if
end procedure
```

V. The Performance Evaluation

To evaluate the effect of the DAO attack on the efficiency of the network and the performance of our proposed mechanism in mitigating that attack, we have conducted a set of experiments using Contiki (Contiki3.0), a lightweight and open-source operating system designed specifically for low-power resource-constrained IoT devices [17]. Contiki features a highly optimized networking stack including several IoT standards such as CoAP, UDP, 6LoWPAN and IPv6. It also features implementations for RPL standard fundamental mechanisms. Cooja [18], a cross-level simulator for Contiki, was used to carry out the simulation experiments, to emulate the exact binary code that runs on real sensor devices. Cooja incorporates an internal hardware emulator called MSPsim [19], which is used in our simulations, to emulate accurately (i.e. impose hardware constraints) the Tmote Sky platform, an MSP430-based board with an ultra-low power IEEE 802.15.4 compliant CC2420 radio chip. We used the Unit Disk Graph Radio Medium (UDGM) radio protocol, the CSMA/CA protocol at the MAC layer and the ContikiMAC as a radio duty cycling (RDC) protocol. The ContikiRPL library was altered to implement the DAO attack on some nodes. In particular, we implemented the attack by means of malicious insider nodes programmed to transmit DAO messages to their preferred parents periodically at preconfigured fixed periods. A set of three malicious nodes running the DAO attack were used. At the application layer, we simulated a periodic data collection application where each node sends one packet to the sink every 60 seconds (the time of sending is randomly chosen within the 60 seconds period). The sink also sends a reply for each received packet to simulate the downward traffic. We have considered in our simulations a uniform distribution where 50 nodes are spread in a square area of 100m x100m. All nodes are static including the DODAG root, which is located outside the square area by a distance of 10 meters. We have selected three nodes at the farthest edge from the root to act as malicious nodes to cover the majority of forwarding paths; this is what attacker might think of to harm the network widely. The number of allowed DAOs forwarded by a parent per child (DAOMax threshold) is set 10 for our proposed mechanism. The rate in seconds at which the attacker sends DAO messages (attack interval) is varied between 0.25 and 10 seconds. For each scenario, five simulation experiments with different seeds were run in order to get statistically valid results. The graphs below show the mean values of the results and the error bars at the 95% confidence interval of the mean. The simulation time was selected to be 1800 virtual seconds for each experiment. The performance evaluation was based the following metrics

Number of DAOs Forwarded: is the average number of forwarded DAOs sent by the parent nodes in the network.

Power Consumption (mW): is the average power consumption at the networks nodes.

Packet Delivery Ratio in the upward direction (Upward PDR): is the average ratio between the number of data packets sent out by the network nodes and the total number of data packets received at the root node.

Packet Delivery Ratio in the downward direction (Downward PDR): is the average ratio between the number of packets received at the nodes and the total number of replies sent out by the root node.

The Upward Latency (seconds): is the average end-to-end delay of all packets sent by the nodes and received successfully at the root.

The Downward Latency (seconds): is the average end-to-end delay of all replies sent by the root and received at the nodes

We have evaluated the performance of RPL, InsecRPL (i.e. RPL under DAO attack), and SecRPL (i.e. RPL under attack with our proposed mitigation mechanism) in terms of previous mentioned metrics. Fig. 1 shows the average number of forwarded DAO messages per node under various attack intervals. The DAOMax threshold is set to 10 per destination. As can be observed in Fig. 1, both InsecRPL and SecRPL have registered a higher overhead in terms of forwarded DAOs compared to the reference model (RPL) which is proportional to the attack interval. However, Fig.1 also shows that SecRPL has registered much less overhead compared to the insecure version especially under heavy attack (attack interval of 250 milliseconds). This also holds true within the case of energy consumption as shown in Fig. 2, which can be attributed to the mechanism of restricting the number of DAOs that can be forwarded by a parent per destination. Indeed, Fig. 2 shows that the InsecRPL has experienced a relatively high power consumption profile, which is much related to the increase in the DAO overhead. In fact, the power consumption profile in ContikiOS is calculated by adding up four components, the idle, listening, transmission and receiving. Hence, the increase in the number of DAOs forwarded increases the power consumed by the forwarder nodes (transmission and receiving components). In addition, it affects the listening time of a forwarders children nodes, though they are not forwarders themselves, (listening component) by forcing them to listen for longer periods due to the congestion at that forwarder node.

The upward and downward latencies of compared protocols are illustrated in Fig. 3 and Fig. 4 respectively. Similarly, it is clear that the DAO attack has an adverse effect on the latency in both directions, which can be attributed again to the congestion induced by the attack at the forwarder nodes. In Fig. 5 and 6, we show the performance of the three protocols in terms of upward PDR and downward PDR respectively. The figures show that mounting the attack with a high attacking interval, affects negatively both the upward and downward traffic patterns (i.e. under this topology, in fact, the effect of attack may differ under different topologies or under different data traffic rates). This can be attributed mainly to the congestion incurred by the increase in the number of forwarded DAOs. This has been mitigated in the proposed solution, which registers PDRs comparable to that of the reference model.

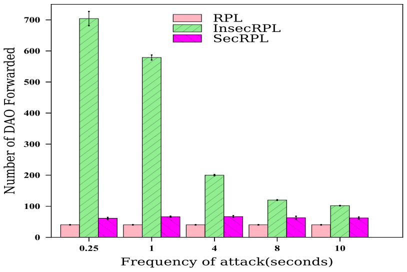

Fig. 1: DAOs forwarding overhead under various attack intervals

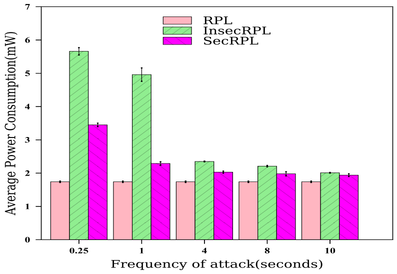

Fig. 2: Average power consumption under different attack intervals

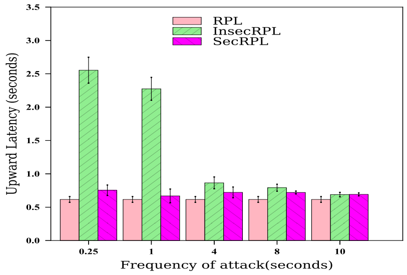

Fig. 3: The upward latency under various attack intervals

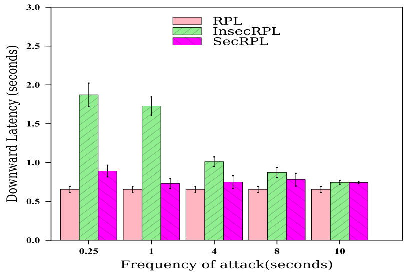

Fig. 4: The downward latency under various attack intervals

A. The Effect of the Threshold Parameter (DAOMax)

Another point we study here is the effect of our mitigation mechanism on the reliability of networks in terms of packet delivery ratio. It is clear that setting the threshold value to a small number will minimize the energy consumption and control overhead but at the cost of reliability. This is illustrated in Figs. 5 and 6. Fig. 5 shows that setting the DAO threshold Max to a very small value reduces both the energy consumption and control traffic overhead. However, as illustrated in Fig. 6, this results in a lower downward PDR for any threshold less than four while

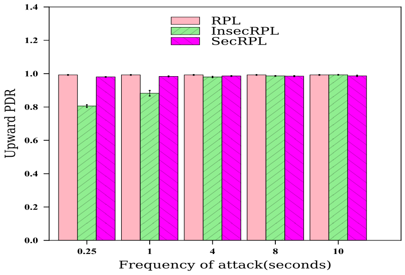

Fig. 5: The upward PDR under different attack rates

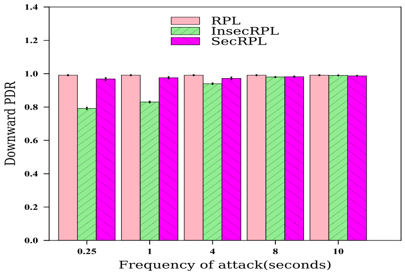

Fig. 6: The downward PDR under different attack rates

the upward PDR is not affected. This indicates that setting the DAOMax to a small value negatively affects only the downward traffic. In fact, setting the DAOMax to a small value will prevent the intermediate parent nodes from forwarding some critical DAO messages necessary to build downward routes, thus explaining the lower downward PDR.

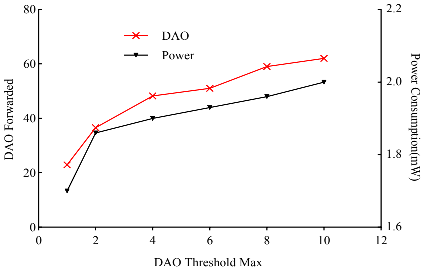

Fig. 7: Power consumption and control overhead under various thresholds

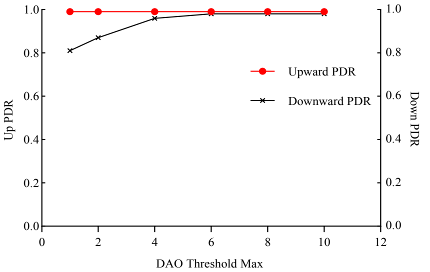

Fig. 8: Downward and upward PDRs under various thresholds

VI. Conclusion

In this article, we have presented the DAO attack, which is triggered by having a malicious node send DAO control messages to its parent. This attack differs from other hello-based exploits (such as DIS and DIO attacks) since DAO messages are transmitted in end-to-end fashion (i.e. from the sensor node to the root). Thus, the level of damage is not restricted to the local scope of the attacker. In fact, a DAO message sent by a child node located on the edge of the network will trigger network-wide DAO transmissions since DAO messages are forwarded by every intermediate parent between that child and the DODAG root. In addition, this kind of attack can be mounted simply without the need to compromise security keys from legitimate nodes. We have shown how this attack may significantly harm the performance of the network especially in terms of power consumption and reliability. Our experiments illustrate that DAO attacks significantly increase the control traffic overhead and power consumption while moderately affecting downward traffic reliability under the chosen assumptions. We have, further, proposed and assessed a mechanism to mitigate the effect of such an attack.

References

[1] J. Hui and D. E. Culler, ”Extending IP to Low-Power, Wireless Personal Area Networks, ” in IEEE Internet Computing, vol. 12, no. 4, pp. 37-45, 2008.

[2] J. Hui, P. Thubert, ”Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks,” RFC 6282, September 2011.

[3] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, K. Pister, R. Struik, J.P. Vasseur, R. Alexander, ”RPL: IPv6 routing protocol for low-power and lossy networks,” RFC6550, March 2012.

[4] T. Clausen, U. Herberg and M. Philipp, ”A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL),” in the 7the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Wuhan, 2011, pp. 365-372.

[5] IETF ROLL Working Group, ”Charter for Working Group,” [Online]. Available: https://datatracker.ietf.org/wg/roll/charter. [Accessed: 23- April- 2018].

[6] A. Dvir, T. Holczer and L. Buttyan, ”VeRA - Version Number and Rank Authentication in RPL,” in the 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, Valencia, 2011, pp. 709-714.

[7] L. Wallgren, S. Raza, and T. Voigt, ”Routing Attacks and Countermeasures in the RPL-Based Internet of Things,” in International Journal of Distributed Sensor Networks, vol. 9, no. 8, 2013.

[8] M. Landsmann, M. Wahlisch and T. C. Schmidt, ”Topology Authentication in RPL,” in the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Turin, 2013, pp. 73-7.

[9] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schnwlder, Mitigation of topological inconsistency attacks in RPL-based lowpower lossy networks,” in the International Journal of Network Management, vol. 25, no. 5, pp. 320-339, 2015.

[10] A. Mayzaud, R. Badonnel and I. Chrisment, ”Detecting version number attacks in RPL-based networks using a distributed monitoring architecture,” in the 12th International Conference on Network and Service Management (CNSM), Montreal, QC, 2016, pp. 127-135.

[11] F. Ahmed, and Y.-B Ko, ”Mitigation of black hole attacks in Routing Protocol for Low Power and Lossy Networks”, in Security and Communication Networks, 9: 5143-5154, 2016.

[12] A. Aris, S. F. Oktug and S. Berna Ors Yalcin, ”RPL version number attacks: In-depth study,” in the IEEE/IFIP Network Operations and Management Symposium (NOMS), Istanbul, 2016, pp. 776-779.

[13] D. Airehrour, J. Gutierrez, and S. K. Ray, ”Secure routing for Internet of Things: A survey,” in Journal of Network and Computer Applications, vol. 66, pp. 198-213, May 2016.

[14] P. Perazzo, C. Vallati, G. Anastasi and G. Dini, ”DIO Suppression Attack Against Routing in the Internet of Things,” in IEEE Communications Letters, vol. 21, no. 11, pp. 2524-2527, November 2017.

[15] U. Herberg and T. Clausen, ”A comparative performance study of the routing protocols LOAD and RPL with bi-directional traffic in lowpower and lossy networks (LLN),” in Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, 2011, pp. 73-80.

[16] A. Dunkels et al., ”Contiki: The Open Source OS for the Internet of Things,” [Online]. Available: http://www.contiki-os.org. [Accessed: 17- May- 2018]

[17] A. Dunkels, B. Gronvall and T. Voigt, ”Contiki - a lightweight and flexible operating system for tiny networked sensors,” in the 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA, 2004, pp. 455-462.

[18] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne and T. Voigt, ”Cross-Level Sensor Network Simulation with COOJA,” in Proceedings of the 31st IEEE Conference on Local Computer Networks, Tampa, FL, USA, 2006, pp. 641-648.

[19] J. Eriksson, A. Dunkels, N. Finne, F. Osterlind, and T. Voigt, ”Mspsim an extensible simulator for msp430-equipped sensor boards,” in Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Delft, The Netherlands, Poster/Demo Session, January 2007.