Specification-based IDS for securing RPL from topology attacks

Anhtuan Le, Jonathan Loo, Yuan Luo, Aboubaker Lasebae
Middlesex University
London, United Kingdom
{a.le, j.loo, y.luo, a.lasebae}@mdx.ac.uk

***Abstract*—This paper focuses on the security aspect of RPL (Routing Protocol for Low-power and lossy network) by introducing a new type of threat – the topology attack, which changes the node operation for breaking the optimised network topology, and designing a specification-based IDS for detecting it. We present two novel RPL attacks of this type: the rank attack and local repair attack. We also propose an IDS architecture using network monitor backbone, and describe its monitoring mechanisms through a RPL finite state machine implemented in each monitor node. We show that our system can effectively detect these routing operation threats with a reasonable overhead.**

Keywords: RPL; topology attack; rank attack; local repair attack; IDS; specification-based

I. INTRODUCTION

RPL is an underlying protocol for 6LoWPAN, an IETF promising standard to bring the ubiquitous ideal vision to real life. Maintaining a reasonable performance for RPL is a crucial issue for making this standard to be public accepted. However, 6LoWPAN devices have weak secured nature, and the network suffers from many routing security threats coming from both the external and internal attackers. There are few proposed solutions for RPL security, and most of them focus on using cryptography to secure the RPL control messages. Cryptography solutions, nevertheless, cannot protect the network from internal attackers if the encryption keys are compromised. Internal attackers can control the communication and downgrade the network performance by using the compromised nodes. Intrusion detection system (IDS) is an effective approach for monitoring network behavior for early detecting those malicious behaviours. The three most widely used approaches in IDS are misused, anomaly-based and specification-based. Misuse solutions needs to define attack signatures, so they are not favored in RPL security because the attacks in this environment are not well-defined. Anomaly-based approaches are based on monitoring the node performance to define a threshold for differentiating compromised or benign nodes. However, working with maximum performance does not necessarily mean the genuine behaviours, because the compromised nodes can break the optimized topology first then work like normal and still downgrade network performance. There is a kind of attack that changes the node operation to not follow the routing protocol to create bad topology. Anomaly-based solutions will fail on detecting such kind because they do not consider the node operations. Specification-based IDS is the only suitable solution for monitoring inside the node operations to guarantee that this node follows all the routing rules and provides an optimized topology.

In this paper, we introduce a new kind of attack that damages the optimal network topology by breaking protocol operations. We present two novel routing operation attacks in RPL: the rank and local repair attack. We then design a network monitoring architecture and a RPL specification-based IDS with a finite state machine for malicious checking in each monitor node. This is, to the best of our knowledge, the first specification-based IDS for RPL. We show that our solution can detect the RPL routing operation threats, and consume only a reasonable overhead. The rest of the paper is organized as following: Section II presents some background of RPL and its topology attacks while Section III reviews the security countermeasures and Section IV introduces our solution design. Section V provides some evaluation analysis about the detection ability of the system whereas Section VI concludes the paper.

II. RPL TOPOLOGY ATTACKS

The RPL architecture is a combination of multiple Destination Oriented Directed Acyclic Graphs (DODAG) networks, each of these can be considered like many wireless sensor devices connected to a DODAG root. Those roots are connected together and to the Internet through a backbone or transit link. The main RPL focus is to make the routing topology to be auto-optimized, while prevent any loops from happening [1]. The loop prevention mechanism is based on the Rank concept to show the node relationship. Each node needs to compute a rank which based on collected information from its neighbours. Every node except the sink needs to choose a preferred parent, and the rank of a parent must always not be larger than the rank of its children. The auto-optimised topology is maintained by the local and global repair mechanism which will fix any broken link.

Since RPL devices have the weak secure nature without the tamper-resistant ability, attackers can capture the node, extract all the cryptography information and utilize it for working legally in the network. Once capturing the nodes, attackers can also implement malicious code inside to break some routing operation rules. This kind of changing is difficult to detect because the inside processing of a node is only checked by itself. Its neighbours unaware of the change and if the protocol is continued to process while some nodes do not follow its rules, the optimized topology can be broken. Attacks are even more dangerous and difficult to be detected when the malicious nodes cooperated. RPL is vulnerable to this kind of attack because it has many strict rules to help to maintain the optimized state. In the following part, we analyze two examples of the topology attacks that an adverse may utilize to interfere with network performance: the rank attack and the local repairing attack. Both of them aim at changing the inside operation of the nodes for breaking the optimized network topology.

A. Rank attack

RPL has a strict rule about the node rank that "rank strictly increases in the Downstream direction and strictly decreases in the Upstream direction". Considering a scenario when the source - node 1 sends the packet to the destination - node N through intermediate nodes 2, 3, 4, …, N-1. Assume the rank of these N nodes are R_1, R_2, R_3, …, R_{n-1}, R_n consequently. The rank rule states that if node 1 sends packets upward to node N then the condition $R_1 \geq R_2 \geq R_3 \geq \ldots R_{n-1} \geq R_n$ must be satisfied; or if the route is downward then $R_1 \leq R_2 \leq R_3 \leq \ldots R_{n-1} \leq R_n$ must be satisfied. The senders and receivers along the route have the responsibility to check these conditions and inform any broken of this rule by setting the Rank-Error bit in the RPL Packet Information [2]. The rank attack is easy to be implemented by simply skipping the rank checking function in the compromised nodes. This attack is difficult to be revealed because it does not need to spoof anything, and most of the behaviours of the compromised nodes look like normal from their neighbours' point of view. Once the rank rule is broken, the consequence can be (i) un-optimized path is created (ii) if the attack is initiated in the route discovery phase, some optimized paths may be disrupted, which mean they exist but will never be discovered, and (iii) a loop can be created without any detection. These consequences definitely downgrade the network performance in many important Quality of Service aspects, such as throughput and delay.

B. Local repair attack

A node in RPL can start the local repair progress in two following ways [2]: The first way is the poisoning mechanism by changing its rank to infinitive and broadcast this rank to all of its neighbours. Those neighbours once receive and update the rank information of that node may need to find a new parent towards the root. The second way to do local repair is to change DODAG ID value of the node. This metric is unique to each DODAG and show what LoWPAN the node belongs to. A node changes its DODAG ID means that it left that DODAG and now belongs to a new DODAG neighbour. As a result, all of its child nodes need to do a local repair to find for a new preferred parent. In RPL, the node is supposed to only do local repair if the links towards its parent list are all broken. However, the adverse can make the node change its DODAG ID or broadcast infinitive rank frequently without any reason. Only the node itself can verify the state of the link to its preferred parent, so when the other neighbours look at a frequently local repair made by a node, they cannot justify whether that node is benign or not. Every time a local repair happens, network topology will need to be updated. This will cost resources and degrade network operation. In case of a node changing its DODAG ID, it is even worse because that node can create local repair in at least two DODAG.

III. RELATED SECURITY COUNTERMEASURES

There are not many works presented on securing RPL. IETF RPL specification [2] proposed mechanisms for securing RPL control messages such as DODAG Information Objective (DIO), Destination Advertisement Object (DAO), DODAG Information Solicitation (DIS). This solution uses symmetric key and public key cryptography to secure the control messages but not consider the establishment and maintenance of the keys. Tsao [3] specified the normal behaviours of RPL control messages such as DIO, DAO, DIS and the control information place in the user data flow of IPv6 Flow Label. IDS is an essential approach for monitoring and preventing RPL from internal attacks. However, to the best of our knowledge, there is no proposed IDS for RPL. Specification-based approach is the only direction that can detect the threats by profiling and monitoring node operations. The main techniques of specification-based IDS are finite state machine transitions, machine learning for pattern recognition and statistical analysis to derive automatically the program specifications [4, 5]. Literature also presented specifications-based IDS on some protocols working in a similar environment with 6LoWPAN such as AODV [6], OLSR [7] and CoP [8].

IV. SPECIFICATION-BASED IDS FOR RPL

A. Monitoring architecture

We assume that during the setting up of the network, a back bone of monitor nodes (MN) is also created and satisfy the following requirements (i) the number of MNs should be minimal (ii) all the MNs are trusted and have enough ability to perform the additional monitor works (iii) that backbone can cover the whole network, which mean every node in the network is in the range of at least one MN. After that backbone is set up, a monitor node will sniff the communication from its neighbours, which includes its parent and child nodes. MN will make an entry for each of its neighbours to store monitoring data for that node. The monitoring data are (i) Object ID and its rank (ii) Preferred parent ID and its rank (iii) Number of topology change/set up in a period of time. All of these are monitored and updated by analysing the DIO messages from the object. When MN discovers a node working maliciously, but it cannot decide whether that node is an attacker, it can request other MNs to collect more information for decision support. Figure 1 shows an example of this architecture. Node 9 and 13 are the two MNs. Node 13 cover node 1, 2, 3, 5, 6, in which node 5 is its preferred parent, node 2 is its child while 1, 3, 6 is its neighbours. Node 6 is double monitored by MN 9 and 13 and its information can be cross checked.

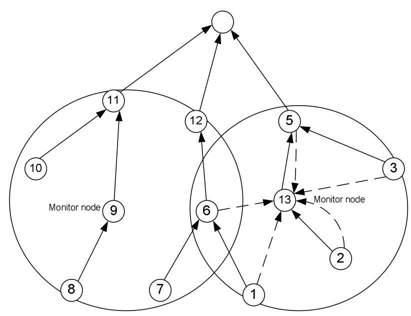

Figure 1. Monitoring node architecture

This design only requires a reasonable overhead because most of the overhead is from the set up phase, which cost only one time per network lifetime, or from the cross-checking, which is optional and only be raised if the possibility that a threat happens is high. The monitor node only sniffs transmission among its neighbours so it does not add any more communication overhead.

B. Finite State Machine for RPL

A finite state machine will be implemented in each monitor node for monitoring the behaviour flow of the object. The state machine can be shown in figure 2. There are four main normal states: the start when monitored object join the network, the topology setup/change, the sending and receiving control messages. When a monitor node first hears a DIO message from the object, its FSM will move from the start to the topology setup/change state. The monitor node then extracts all the necessary information from that DIO in a specific entry for the object in its monitoring table. From the topology change/setup state, depend on hearing the object sending or receiving control messages, FSM moves to sending or receiving state respectively. If FSM detects any change in the DIO message related to the preferred parent, change in the DODAG ID or the rank goes to infinitive, it will move back to the topology setup/change state.

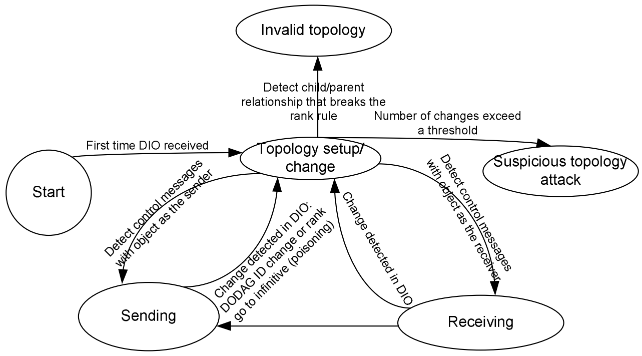

Figure 2. Finite State Machine for RPL operation

FSM has two states to indicate the malicious behaviours of the routing operations, which come from the topology setup/change state. When checking rank information in the received DIO, if a monitor node detects any child/parent relationship that breaks the rank rule, FSM will move to the invalid topology state. If the FSM goes to the topology setup/change more frequently so that the number of changing exceeds a threshold, it will go to the suspicious topology attack, which assume that the operation of the monitored node break the stable of network topology.

V. DETECTING THE ATTACKS

A. Rank attack detection

We consider a scenario when two malicious nodes are cooperated to break the rank rule. With our monitoring system, there should be a monitor node that covers the malicious node with lower rank. This monitored object needs to provide its information regarding its rank and the parent's rank. If none of these two nodes forge the rank information, then the monitor node will reveal a fact that the rank of the parent is higher than the rank of the child, which breaks the rank rule. Rank attack is therefore detected. On the other hand, if the malicious nodes forge their rank information and use those fake ranks, there is no breaking in the rank rule, however, because the malicious nodes change between their real rank and forged ranks, the monitoring system can capture this change and suspect about their behaviours. The monitor node then can start cross checking to reveal the attackers.

B. Local repair attack

Any node behaviour that leads to a local repair will be recorded in the monitor node. The number of the local repair that each object caused is calculated from this information. If there are too many local repairs that exceed a threshold then the monitoring system will raise an alarm for local repair attack.

VI. CONCLUSION

In this paper, we discuss the RPL topology attacks by breaking node operations. We introduce two new attacks of this type: the rank and local repair attack. We also propose a specification-based IDS with finite state machine design to prevent those threats. The idea of a specification-based IDS is to build manually an abstract of the normal network operation, and detect the malicious behaviours that break those specifications. This research is the first attempt in specifying the RPL operation in order to protect the routing operation attacks. We design the architecture of a monitoring system for RPL and the information that it should collect to analyse. We show that the system can detect effectively these RPL topology attacks with a reasonable overhead. Our next target is to implement this system in simulation environments such as Contiki, and analysis its effectiveness. We also interested in expanding and improving the FSM to develop a more robust specification-based IDS for that protocol.

REFERENCE

[1] Vasseur, J.-P. and A. Dunkels, *Interconnecting Smart Objects with IP: The Next Internet*. 2010, Morgan Kaufmann.

[2] Winter, T., et al. (2011) *RPL: IPv6 Routing Protocol for Low power and Lossy Networks - draft-ietf-roll-rpl-19: http://tools.ietf.org/html/draft-ietf-roll-rpl-19*.

[3] Tsao, T. (2010) *Internet-Draft v00: A Security Design for RPL: IPv6 Routing Protocol for Low Power and Lossy Networks http://tools.ietf.org/html/draft-sdt-roll-rpl-security-00*.

[4] Sekar, R., et al., *Specification-based anomaly detection: a new approach for detecting network intrusions*, in *Proceedings of the 9th ACM conference on Computer and communications security*. 2002, ACM: Washington, DC, USA. p. 265-274.

[5] Stakhanova, N., S. Basu, and J. Wong, *On the symbiosis of specification-based and anomaly-based detection*, in *Computers & security 29 (2010)*. 2010. p. 253 – 268.

[6] Grönkvist, J., A. Hansson, and M. Sköld, *Evaluation of a Specification-Based Intrusion Detection System for AODV*, in *The Sixth Annual Mediterranean Ad Hoc Networking WorkShop*. 2007: Corfu, Greece. p. 121-128.

[7] Tseng, C.H., et al., *A Specification-Based Intrusion Detection Model for* OLSR, in *Recent Advance in Intrusion Detection RAID 2005*. 2005. p. 330-350.

[8] Mostarda, L. and A. Navarra, *Distributed Intrusion Detection Systems for Enhancing Security in Mobile Wireless Sensor Networks.* International Journal of Distributed Sensor Networks, 2008. **4**: p. 83–109.