Demo: Topological Robustness of RPL with TRAIL

Martin Landsmann*, Peter Kietzmann*, Thomas C. Schmidt*, and Matthias Wählisch†
* iNET RG, HAW Hamburg – Berliner Tor 7, 20099 Hamburg, Germany
† CSL, FU Berlin – Takustr. 9, 14195 Berlin, Germany
* {martin.landsmann,peter.kietzmann,t.schmidt}@haw-hamburg.de
† m.waehlisch@fu-berlin.de

Abstract

Dependability in IoT networks largely relies on the robustness of a routing protocol such as *Routing Protocol for Low-power and Lossy Networks* (RPL). Routing is expected to be secure and resilient, in particular against adversaries who try to break into the topology or disturb the overall operation of the network. In this demonstration, we will showcase *Trust Anchor Interconnection Loop* (TRAIL)—our approach to securing RPL that successfully prevents attacks on the topology and isolates attacking nodes.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols, Security*

Keywords

IoT, routing security, mobile security, topology verification

1 Introduction

The hierarchical composition of a RPL [5] topology binds child nodes to a parent node, while each node calculates its topological position, based on the individual *rank* inherited from its parent-node. RPL does not define mechanisms to secure and verify the correctness of an announced rank, which renders it being vulnerable to rank forgery and rank spoofing attacks. Using these weaknesses, an adversary can pretend to be in an arbitrary topological position. This allows him to cause a partition of the topology which in the worst case has impact on a greater number of nodes. As result of such an attack, the adversary attracts a large amount of traffic facilitating further attacks on the passing information and payload. To cope with this vulnerability in RPL, we developed TRAIL [3], a lightweight security approach for the IoT. TRAIL allows to validate the complete path from any node to the root of a RPL network with a single traversing message.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2016
15–17 February, Graz, Austria
© 2016 Copyright is held by the authors.
Permission is granted for indexing in the ACM Digital Library
ISBN: 978-0-9949886-0-7

In this demonstration, we address the vulnerabilities of RPL (§2) and show the design principles of the TRAIL security measurements (§3). Practically (§4) we show (i) a successful attack on RPL causing the reconstruction of large parts of the topology, and (ii) RPL with applied TRAIL that successfully preventing the attack and isolating the attacker.

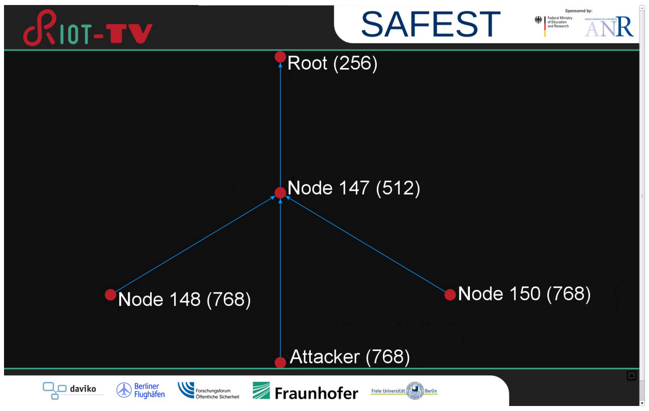
2 Vulnerabilities of RPL

The RPL protocol constructs *destination oriented directed acyclic graph* (DODAG), i.e., trees with a single root-node. Each node in a DODAG has a rank calculated from its relationship to a parent-node. This rank logically describes the topological distance to the DODAG root. To join a DODAG, a node chooses a parent-node from the existing DODAG and computes its own rank by increasing the received parent rank. Then it announces its rank to offer being a parent for other joining nodes. Thus, ranks rise monotonically with topological distance to the DODAG root and a low rank indicates being a beneficial parent-node to choose. While RPL forbids being endlessly greedy to prevent oscillation between two nodes, it allows a node to switch to a more beneficial parent and adjust its rank accordingly. Figure 1(a) depicts a healthy DODAG. The red dots are the nodes with their rank placed in brackets. The arrows show the parent-child communication in the DODAG.

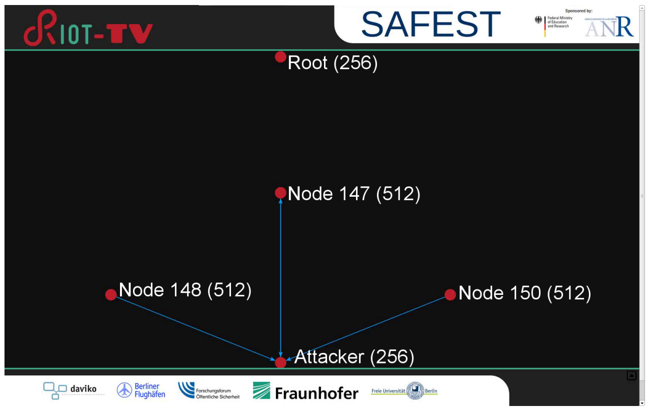
An adversary can exploit the rank based construction and maintenance of a DODAG. By announcing a beneficial low rank, neighbor-nodes will choose the adversary as their new parent. In turn they will recalculate and announce their new lower rank too, attracting their neighbors. This cascades throughout the DODAG until the deceived nodes and the remaining unaffected nodes are in equilibrium. Exploiting this weakness, enables to pull a major amount of traffic towards an attacker to apply *sinkhole* attacks [4]. Periodical switching its rank and triggering DODAG maintenance operations on numerous nodes, enables to disturb or even interrupt a RPL topology. Figure 1(b) shows the DODAG after *Attacker* announced a low rank and successfully deceived its neighbors.

3 TRAIL in a nutshell

A node that requires validation of a complete path to the root node creates a TRAIL validation message extending it with a nonce using Bloom filter [2] compression. The TRAIL validation message is forwarded to its parent node. Upon reception, the parent node validates if rank of the child



(a) RPL DODAG before the attack



(b) Partitioned RPL DODAG after the attack

Figure 1. Successful attack on RPL

node is not lower than the own rank. A violation of monotonic rank order indicates either a RPL inconsistency or an attack, causing the parent node to drop the TRAIL validation message. If the monotonic rank order is proven, the parent node creates a Bloom filter from a chosen nonce and prepends it to the TRAIL validation message before forwarding it to its parent node. Upon reception of multiple validation messages, a node validates the monotonic rank order for each child. The node merges all Bloom filters of the TRAIL validation messages aligned on the last prepended filters, i.e., the one inserted by its child nodes, resulting in a new single TRAIL validation message. Note: Bloom filters can be short and correspond to the number of child nodes.

The TRAIL validation message propagating to the root node contains all nonces from one topological distance to the root node in a single Bloom filter. The position of each filter in the message represents the logical distance to the root node. The root node validates the monotonic rank order and merges the TRAIL validation messages. Then it signs the TRAIL validation message and forwards it back to all child nodes. When a child node receives a signed TRAIL validation message, it first verifies the signature and than probes if its nonce is in the compressed Bloom filter at its assumed topological position in the message. On success, the complete path to the root node is trustworthy to forward traffic. The signed TRAIL validation message is than forwarded to all child nodes. Whenever this validation fails the path to the root node is assumed as compromised and the validation message is dropped. Any inconsistency is detected immediately in both direction by the node receiving the message. Since propagation of a compromised TRAIL validation message is stopped, the adversary is directly identified and isolated from the topology.

Path validation using TRAIL ensures a monotonic topology structure where inconsistencies are directly detected. TRAIL successfully protects against rank spoofing attacks, protecting the topology from partitioning. Attacking nodes are reliably identified and isolated from the topology just by the procedure of TRAIL. The characteristics of TRAIL strengthens resilience against attacks and inconsistencies, which is the base for dependable communication in the IoT. A comprehensive analysis of TRAIL can be found in [3].

4 Demo

In our demonstration, we showcase the topological robustness of RPL using TRAIL. For the sensor nodes we use Atmel SAM R21 Xplained Pro boards running RIOT-OS [1]. We present the following 2 demo scenarios:

1. We setup several sensor nodes to build a DODAG with multihop routes using RPL without TRAIL. After the DODAG converges, an arbitrary node starts an attack by announcing a more beneficial rank, i.e., the rank of the root-node. This will result in a partition of the DODAG. Then we activate TRAIL on *all* nodes and monitor healing of the DODAG and isolation of the attacker.

2. We setup all nodes as before, but activate TRAIL before starting RPL. This demonstrates RPL protection with TRAIL during the DODAG bootstrap phase. Then again we choose an attacking node that will announce a beneficial rank. Afterwards we start RPL on *all* nodes to construct a DODAG. We will observe that the attacking node cannot successfully attract child nodes and eventually will remain isolated.

Acknowledgments

This work is funded by the Federal Ministry of Education and Research (BMBF) of Germany in the Project SAFEST.

5 References

[1] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T. C. Schmidt. RIOT OS: Towards an OS for the Internet of Things. In *Proc. of the 32nd IEEE INFOCOM. Poster*, 2013. IEEE Press.

[2] B. H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Commun. ACM*, 13(7):422–426, July 1970.

[3] H. Perrey, M. Landsmann, O. Ugus, M. Wählisch, and T. C. Schmidt. TRAIL: Topology Authentication in RPL. In *Proc. of Intern. Conf. on Embedded Wireless Systems and Networks (EWSN '16)*, New York, NY, USA, Feb. 2016. ACM.

[4] K. Weekly and K. Pister. Evaluating Sinkhole Defense Techniques in RPL Networks. In *Network Protocols (ICNP), 2012 20th IEEE International Conference on*, pages 1–6, Nov. 2012.

[5] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, IETF, March 2012.