ROS-Defender: SDN-based Security Policy Enforcement for Robotic Applications

Sean Rivera, Sofiane Lagraa, Radu State
SnT, University of Luxembourg, Luxembourg
firstname.lastname@uni.lu

Cristina Nita-Rotaru,
Northeastern University
c.nitarotaru@northeastern.edu

Sheila Becker
Defence Directorate, Luxembourg
sheila.becker@mae.etat.lu

securityandtrust.lu

Introduction & Background

Problem

- The Robotic Operating System (ROS) is a popular framework for robotic system development using a publish-subscribe system
- Current solutions for security of ROS do not provide protection against compromised nodes or denial of service and do not support reactive low-level network policies that can be updated dynamically at runtime

Challenges

- ROS is designed around the concept of topics, higher semantic than network packets, users would like to define policies per-topic
- Most attacks happen and get detected on network level
- Need fast detection and response

Opportunities

- SDN offers an opportunity to define and control policies at the network level, thus allow to defend against intrusions

Our Approach

Design

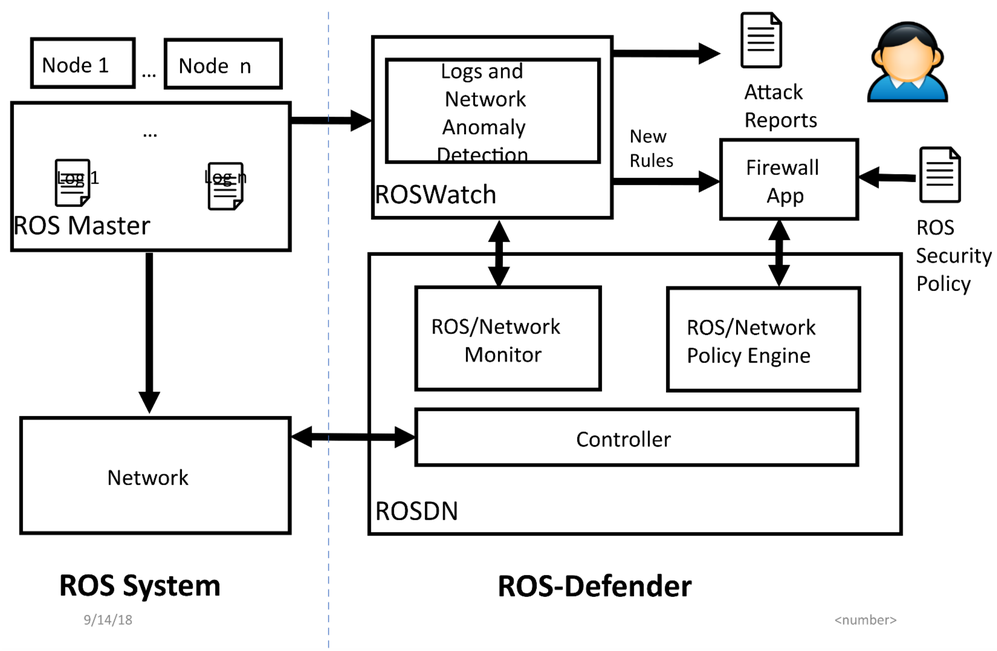
- Create ROS-Defender that leverages SDN to
  - control per-flow policies
  - augment per-flow control, with a per-topic control, to accommodate ROS semantics.
- Define a ROS-Policy-Language that allows a user to specify policies for ROS applications, enforced at network level

We take an intrusion prevention approach where

- a user-defined policy is enforced at network level with the help of ROSDN which replaces the normal ROS network communication with a SDN
- attacks are detected by ROSWatch, a log and network-based anomaly detection which not only detects attacks but also learns rules that can be used to automatically update the policy and prevent future attacks.
- A policy engine ensures that policies expressed in ROS-level abstractions by a user familiar with ROS, are translated into network-level policies.

Advantages

The approach does not require any API changes or software alterations on the part of the developer and it provides security benefits to both ROS and SROS.



Experimental Results

Turtlebot3 robot

This system consists of a Turtlebot3 robot, a Northbound Zodiac Wx switch, and a laptop which is acting as both the SDN controller and ROS master/decision maker.

Virtual robot

This system is a purely virtual version of our Turtlebot3 system. It leverages the Gazebo package (a well-known robotics simulator) to emulate the sensor input for a variety of different of robotic systems, and uses Mininet for the networking component.

- **Turtlebot** - Standard simulated robot for ROS platform
- **Husky** - An outdoor rugged ground vehicle
- **AR Drone** - Simulated Autopilot for the AR Drone platform
- **ABB industrial robot** - Simulated pick and place industrial robot with ROS and Robot Studio
- **Udacity** - Simulator for a self driving car

ROSDN Performance analysis	Avg. Latency	Max BW	Time
RTurtlebot, No SROS or ROSDN	.0164s	190kB/s	46.7s
RTurtlebot, ROSDN	.024s	190kB/s	49.5s
RTurtlebot, SROS	NA	220kB/s	55.108s
RTurtlebot, SROS and ROSDN	NA	220kB/s	68.712s
VTurtlebot, No SROS or ROSDN	0.00426s	230 kB/s	47.1s
VTurtlebot, ROSDN	0.00568s	230 kB/s	57.429s
VTurtlebot, SROS	NA	290 kB/s	61.2s
VTurtlebot, SROS and ROSDN	NA	290 kB/s	75.432

Simulated Analysis	Time	Overhead
Simulated Turtlebot, No ROSDN	42.101	0.0%
Simulated Turtlebot, ROSDN	46.5	10.45%
Simulated Husky, No ROSDN	33.59	0.0%
Simulated Husky, ROSDN	34.6025	3.014%
Simulated AR Drone, No ROSDN	23.68	0.0%
Simulated AR Drone, ROSDN	25.5	7.69%
Simulated ABB industrial robot, No ROSDN	17.62	0.0%
Simulated ABB industrial robot, ROSDN	19.14	8.62%
Simulated Udacity self driving car, No ROSDN	170.5	0.0%
Simulated Udacity self driving car, ROSDN	173.25	1.61%

Conclusion

In this paper we proposed ROS-Defender a, comprehensive security architecture for ROS based robotic systems. ROSDefender does not require changes to the existing ROS code since it's using available ROS application programming interfaces and SDN level mechanisms to monitor and execute access control actions.

UNIVERSITÉ DU LUXEMBOURG