(Application-level) Security for ROS-based Applications


Bernhard Dieber\textsuperscript{1}
bernhard.dieber@joanneum.at

Benjamin Breiling\textsuperscript{1}, Severin Kacianka\textsuperscript{2}, Stefan Rass\textsuperscript{3}, Peter Schartner\textsuperscript{3}

\textsuperscript{1} JOANNEUM RESEARCH, Institute for Robotics and Mechatronics, Austria
\textsuperscript{2} Technical University of Munich, Software Engineering Group, Germany
\textsuperscript{3} Alpen-Adria Universität Klagenfurt, Institute of Applied Informatics, Austria
Issues in industrial security

- Newly evolving vulnerabilities in industrial control networks
  - Internet of Things
  - Teleworking / remote access
  - Cloud computing
  - Bring your own device
  - …

- Several successful attacks have been reported (not always officially)
  - Stuxnet
  - Jeep hack
  - Ukrainian power grid hack
  - Robots manipulated to put less welding points
  - …

- Network security is not enough
  - Multi-layer security is required
Outside industry

- Use of ROS even more likely (service robots, toys, …)
- Use in private homes
- Attacking the network is even simpler
  - Enter via malware, baby monitor, smart meter, ….
  - (Even) lower security awareness
- Higher volume of devices
- Massive privacy issues
Security in ROS

- Decoupling and transparency in publish / subscribe systems causes security issues
  - Not a ROS-specific problem
- Security: not part of the design so far – many vulnerabilities*
- No authentication
- No encryption
- No data integrity

Attack vectors on a ROS-application

- Data injection
  - Unauthorized publishing
  - Inject commands or false data

- Data eavesdropping
  - Unauthorized subscribing
  - Gain insight into production process
  - What about camera images? *Privacy!*

- Denial of Service
  - Shutdown nodes by using same name
  - High-frequency publishing
Application-level Security Enhancements [1]

- **Authentication of talkers/listeners**
  - Registration: challenge-response authentication of legitimate components
    (use certificates)
  - Sending messages: mandatory digital signature
  - Receiving messages: reject if not signed or signature invalid

- **Confidentiality**
  - use topic-key shared between all talkers/listeners
  - illegitimate talkers cannot produce "understandable" messages
  - illegitimate listeners won’t understand what others talk about
Application-level Architecture

P2P Security for ROS [2]

- Integrated to ROS core
- Secure low-level communication using TLS
- Establish trust between nodes on p2p basis
- Modify `ros_comm` package
- Similar to SROS but in roscpp, TCP & UDP

Outlook

- Certificate Management
- Security for robot swarms
- Non-centralized security management
Security by optimism and prayer

Expert

Hoping Nobody Hacks You

O RLY?

@ThePracticalDev