Leveraging Secure Discovery Server in ROS 2

Speaker:
- Eduardo Ponz Segrelles, Project Manager @ eProsima
  eduardoponz@eprosima.com

October 20th, 2023
Agenda

01 Motivation
- Out-of-the-box discovery caveats
- Secure deployments

02 ROS 2 Discovery Server
- Overview
- Configuration options
- Network bandwidth performance

03 SROS 2 Security capabilities
- Security plugins
- Configuration options
- SROS 2 tooling

04 Deployment example
- Introduction
- Demo
Motivation

*Why using ROS 2 Discovery Server & SROS 2*

---

**ROS 2 discovery scalability**

Avoid using multicast based discovery and drastically reduce the discovery related traffic by leveraging Discovery Server.
Motivation

Why using ROS 2 Discovery Server & SROS 2

Secure production deployments

Properly isolated robots and prevent ill-intentioned external actions on ROS 2 distributed applications
ROS 2 Discovery Server

A Fast DDS powered unicast based & centralized discovery mechanism for ROS 2
ROS 2 Discovery Server

Out-of-the-box discovery (SDP)

SDP Node Discovery Phase (PDP)

SDP Topic Discovery Phase (EDP)
ROS 2 Discovery Server

Out-of-the-box discovery (SDP)

No configuration
ROS 2 Nodes discover each other automatically
ROS 2 Discovery Server

Out-of-the-box discovery (SDP)

SDP Node Discovery Phase (PDP)

1. No configuration
   ROS 2 Nodes discover each other automatically

2. Traffic heavy
   Number of packets increases exponentially with number of ROS 2 contexts
**ROS 2 Discovery Server**

*Out-of-the-box discovery (SDP)*

1. **No configuration**
   - ROS 2 Nodes discover each other automatically

2. **Traffic heavy**
   - Number of packets increases exponentially with number of ROS 2 contexts

3. **Multicast based**
   - PDP is based on multicast, which may bring problems on WiFi or managed networks
ROS 2 Discovery Server

Discovery Server mechanism

Client Node_1

Client Node_2

Client Node_3

Client Node_4

Discovery Server
ROS 2 Discovery Server

Discovery Server mechanism

Client Node_1

Client Node_2

Client Node_3

Client Node_4

Discovery Server

Pub_1
ROS 2 Discovery Server

Discovery Server mechanism

Client Node₁

Client Node₂

Client Node₃

Client Node₄

Topic₁

Pub₁

Sub₁

Discovery Server
ROS 2 Discovery Server

Discovery Server mechanism
ROS 2 Discovery Server

Discovery Server mechanism

Client Node₁ → Topic₁ → Discovery Server → Sub₁

Client Node₃ → Pub₁ → Discovery Server → Sub₁

Client Node₄ → Pub₁ → Discovery Server

Client Node₂ → Sub₂ → Pub₂

Topic₂
ROS 2 Discovery Server

Discovery Server mechanism
**ROSW 2 Discovery Server**

*Discovery Server mechanism*

---

1. **Minimal configuration**
   - Clients only need to know where the Discovery Server is located.

---

**Diagram:**

1. **Client Node\textsubscript{1}**
   - Sends **Sub\textsubscript{1}** to **Discovery Server**
   - **Pub\textsubscript{1}** from **Discovery Server**
   - Receives **Pub\textsubscript{1}** from **Client Node\textsubscript{3}**

2. **Client Node\textsubscript{3}**
   - Sends **Pub\textsubscript{1}** to **Discovery Server**
   - **Sub\textsubscript{1}** from **Discovery Server**
   - Receives **Sub\textsubscript{1}** from **Client Node\textsubscript{1}**

3. **Discovery Server**
   - **Sub\textsubscript{1}** from **Client Node\textsubscript{1}**
   - **Pub\textsubscript{1}** from **Client Node\textsubscript{3}**
   - **Sub\textsubscript{2}** from **Client Node\textsubscript{2}**
   - **Pub\textsubscript{2}** from **Client Node\textsubscript{4}**

4. **Client Node\textsubscript{2}**
   - Sends **Sub\textsubscript{2}** to **Discovery Server**
   - **Pub\textsubscript{2}** from **Discovery Server**
   - Receives **Pub\textsubscript{2}** from **Client Node\textsubscript{4}**

5. **Client Node\textsubscript{4}**
   - Sends **Pub\textsubscript{2}** to **Discovery Server**
   - **Sub\textsubscript{2}** from **Discovery Server**
   - Receives **Sub\textsubscript{2}** from **Client Node\textsubscript{2}**
**Discovery Server mechanism**

1. **Minimal configuration**
   Clients only need to know where the Discovery Server is located.

2. **Traffic reduction**
   Benchmarks show up to a 85% traffic reduction when compared to SDP.
ROS 2 Discovery Server

Discovery Server mechanism

1. Minimal configuration
   Clients only need to know where the Discovery Server is located.

2. Traffic reduction
   Benchmarks show up to a 85% traffic reduction when compared to SDP.

3. Unicast based
   Works out-of-the-box in WiFi and managed networks as it does not require multicast.
ROS 2 Discovery Server

Server redundancy

Client Node

Client Node

Client Node

Discovery Server

Discovery Server
ROS 2 Discovery Server

Discovery Server deployment and configuration

**Server redundancy**

- Client Node
- Client Node
- Client Node
- Discovery Server
- Discovery Server

**LAN Segmentation**

- Client Node
- Discovery Server
- Client Node
ROS 2 Discovery Server

Discovery Server deployment and configuration

Server redundancy

Bridging segments
Deploy a Discovery Server

Instantiating a Discover Server is as simple as running one CLI command

```bash
$ fastdds discovery -i 0
```
Deploy a Discovery Server
Instantiating a Discover Server is as simple as running one CLI command

```
$ fastdds discovery -i 0
```

Configure nodes as Clients
Nodes are configured as Clients using an environment variable

```
$ export ROS_DISCOVERY_SERVER="192.168.1.54"
```
ROS 2 Discovery Server

Discovery Server deployment and configuration

1. Deploy a Discovery Server
   Instantiating a Discover Server is as simple as running one CLI command
   
   `$ fastdds discovery -i 0`

2. Configure nodes as Clients
   Nodes are configured as Clients using an environment variable
   
   `$ export ROS_DISCOVERY_SERVER="192.168.1.54"`

3. Advanced configurations
   Update list of Server in run-time, Super Client, etc.
ROS 2 infrastructure to leverage DDS Security capabilities and protect your ROS 2 applications
Authentication
(DDS:Auth:PKI-DH)

Authenticate a new Participant when joining the network
DDS Security

Several levels of protection

Authentication
(DDS:Auth:PKI-DH)
Authenticate a new Participant when joining the network

Access Control
(DDS:Access:Permissions)
Limit the access and permissions for the Participants in the network
DDS Security

Several levels of protection

Authentication (DDS:Auth:PKI-DH)
Authenticate a new Participant when joining the network

Access Control (DDS:Access:Permissions)
Limit the access and permissions for the Participants in the network

Encryption (DDS:Crypto:AES-GCM-GMAC)
Encrypt the messages between Endpoints
Root CA
- private key
- certificate
DDS Security
Understanding the security infrastructure

Root CA
- private key
- certificate

Identity CA
- private key
- certificate
DDS Security

Understanding the security infrastructure

Root CA
- private key
- certificate

Identity CA
- private key
- certificate

App Identity
- private key
- certificate
DDS Security

Understanding the security infrastructure

- Root CA
  - private key
  - certificate

- Identity CA
  - private key
  - certificate

- App Identity
  - private key
  - certificate

Used by nodes for authentication & encryption/decryption
Understanding the security infrastructure

Root CA
- private key
- certificate

Identity CA
- private key
- certificate

Permissions CA
- private key
- certificate

App Identity
- private key
- certificate

Used by nodes for authentication & encryption/decryption
DDS Security

Understanding the security infrastructure

- Root CA
  - private key
  - certificate

- Identity CA
  - private key
  - certificate

- Permissions CA
  - private key
  - certificate

- App Identity
  - private key
  - certificate

- Governance policy

Used by nodes for authentication & encryption/decryption
DDS Security

Understanding the security infrastructure

- **Root CA**
  - private key
  - certificate

- **Identity CA**
  - private key
  - certificate

- **Permissions CA**
  - private key
  - certificate

- **App Identity**
  - private key
  - certificate

- **Governance policy**

  Used by nodes for authentication & encryption/decryption

  Encryption policies & Domain-wide rules
DDS Security
Understanding the security infrastructure

Root CA
- private key
- certificate

Identity CA
- private key
- certificate

Permissions CA
- private key
- certificate

App Identity
- private key
- certificate

Governance policy

Permissions policy

Encryption policies & Domain-wide rules

Used by nodes for authentication & encryption/decryption
Understanding the security infrastructure

- **Root CA**
  - private key
  - certificate

- **Identity CA**
  - private key
  - certificate

- **Permissions CA**
  - private key
  - certificate

- **App Identity**
  - private key
  - certificate

- **Governance policy**

- **Permissions policy**

- **Encryption policies & Domain-wide rules**

- **Node-specific topic access control policies**

- **Used by nodes for authentication & encryption/decryption**
Create a keystore
Contains public and private certificates and keys, as well as enclaves

$ ros2 security create_keystore <store>
SROS 2 tooling

SROS 2 CLI for easily enable security in ROS 2

1. **Create a keystore**
   - Contains public and private certificates and keys, as well as enclaves
   ```
   $ ros2 security create_keystore <store>
   ```

2. **Create enclaves**
   - Enclave specific certificates and keys, and CA's public certificates
   ```
   $ ros2 security create_enclave <store> <enclave>
   ```
SROS 2 tooling

SROS 2 CLI for easily enable security in ROS 2

1. Create a keystore
   Contains public and private certificates and keys, as well as enclaves

   $ ros2 security create_keystore <store>

2. Create enclaves
   Enclave specific certificates and keys, and CA's public certificates

   $ ros2 security create_enclave <store> <enclave>

3. Configure nodes for SROS 2
   Command ROS 2 nodes to use specific enclaves for authentication, access control, and encryption

   $ export ROS_SECURITY_KEYSTORE=<store>
   $ export ROS_SECURITY_ENABLE=true
   $ export ROS_SECURITY_STRATEGY=Enforce
   $ ros2 run <pkg> <node> --ros-args --enclave <enclave>
SROS 2 & Discovery Server

Secure Discovery Server configuration & demonstration
# secure_discovery_server.xml

- Discovery Server IP address and port (i.e.: 0.0.0.0:11811)
- Identity CA certificate
- Identity certificate
- Identity private key
- Permissions certificate
- Governance file (signed)
- Permission file (signed)

$ fastdds discovery -i 0 -x <config_xml>