Discovery

Discovery Agents

ActiveMQ uses an abstraction called a Discovery Agent to detect remote services such as remote brokers. We can use discovery for JMS clients to auto-detect a Message Broker to connect to, or to provide Networks of Brokers

There are currently two kinds of discovery agent.

Multicast

The Discovery transport uses our own Multicast based discovery agent to locate the list of URIs to connect to.

For more information see the Discovery Transport Reference.

Zeroconf

ZeroConf is a standard discovery specification that uses UDP / multicast to discovery devices. Its used by Apple's Rendezvous services.
We use the jmDNS project to implement the Zeroconf specification to detect services. This means other Zeroconf
based tools can be used in conjunction with this discovery agent.

To configure discovery in a Broker you should use the Xml Configuration. Here is an example of using discovery to create Networks of Brokers.

If you have one or more brokers running with Zeroconf discovery enabled you can connect to a broker using the brokerURL

```
zeroconf:_activemq_development.
```

This will use Zeroconf to find an available broker and one will be randomly chosen & things will auto-failover on disconnect if there are several brokers running.

LDAP Discovery

ActiveMQ supports the use of LDAP for discovery of brokers.

Please see LDAP Broker Discovery Mechanism for more details.

Trying out discovery

If you run the following commands in separate shells you'll have 2 brokers auto-discovering themselves and 2 clients using fixed-URLs

```
maven -o server -Dconfig=src/test/org/activemq/usecases/receiver-zeroconf.xml
maven -o server -Dconfig=src/test/org/activemq/usecases/sender-zeroconf.xml
maven -o consumer -Durl=tcp://localhost:62002
maven -o producer -Durl=tcp://localhost:62001
```

If you want the clients to use discovery to find brokers, run either of the two 'server' statements above (or both) then run the producer/consumer as follows

```
maven -o consumer -Durl=zeroconf:_activemq.broker.development.
maven -o producer  -Durl=zeroconf:_activemq.broker.development.
```

The transport URL is of the format

```
    zeroconf:<serviceName>
```

where *<serviceName>* is the Zeroconf service name; which seems to start with an underscore (_) and must end with a dot (.). So we can use this service name to distinguish development, UAT & production brokers - or group them into domains etc.

Discovery and Security

When using auto discovery of brokers an attacker may be able to present itself as a legitimate broker and by this way catch and / or manipulate all messages that run over it.

Are there security settings in auto discovery to avoid this?