politiker2ceof

telmich

2007-11-12 v0.1-train

Contents

1 Introduction

This document specifies the commands send from politiker to and from ceof[1] to the politiker.

1.1 Changelog

1.1.1 none

- -

[1] the central EOF-1 application

2 Connection

The politiker is started by ceof at startup and communicates through stdin and stdout.

3 Commands

All commands are send as uint32_t types. **Politiker commands** always begin with **3** ("'3042"' for instance), **answers** or **notifications** from ceof begin with **1** ("'1023"' for instance). After each command follows individual data. The second byte indicates the type of message:

- **30**: politiker messages
  - **300**: (De-)Initialisation
  - **301**: Peer related messages
  - **302**: Message related messages
- **10**: ceof messages
  - **100**: (De-)Initialisation
  - **101**: Peer related messages
  - **102**: Message related messages
  - **103**: Message related answers

3.1 30: Politiker messages

3.2 3000: Register politker

After the "'3000"' the politker directly appends an *uint32_t* containing the version of the politker to ceof protocol it speaks. This specification uses version number "'0"'. Answers from ceof:

- **1100**: sucess, you are connected
- **1200**: version not supported

3.3 3001: Deregister politiker

The politker has some problem and has to exit. Ceof will restart a new instance of it. Answers from ceof:

- none

3.4 3010: Retrieve random peer address and fingerprint

The politker needs some peer information to be used as a hop. Ceof forwards that request to **pmg**[2] and returns the answer to the politiker. Answers from ceof:

- 1010: Data follows
  - *peer_address*: 128 Bytes, 0 padded, 0 terminated
  - *peer_fingerprint*: 40 Bytes char array

3.5 3011: Retrieve number of available peers

The politker needs to know how much "'unique"' peers are available, so it can match the required minimum. Ceof forwards that request to **pmg** and returns the answer to the politiker. Answers from ceof:

- 1010: Data follows
  - *number_of_peers*: uint32_t

3.6 3020: Created an encrypted packet

Passes the following information to ceof:

- The length of the packet (uint32_t) (*pck_len*)
- The packet (*pck*)

After the **politiker** send *pck_len*, **ceof** must respond with either

- **1030**: pagket length is accepted
- **1031**: pagket length is too long

If the response is *1030*, **politiker** should send the pcaket, otherwise this session is finished and **politiker** should drop the packet.

[2] peer manager

3.7 10: Ceof messages

3.8 1020: Create encrypted packet

Passes the following information to the politiker:

- GPG-Fingerprint of the peer (40 Bytes char array) (*fpr*)
- Adress of the peer (128 Bytes, 0 padded, 0 terminated) (*address*)
- The length of the message (uint32_t) (*msg_len*)
- The message (*msg*)

After **ceof** send *msg_len*, politiker must respond with either

- **3020**: message length is accepted
- **3021**: message length is too long

If the response is *3020*, **ceof** should send the messsage, otherwise this session is finished and the next thing the politiker expects is a command.