Hewlett Packard
Enterprise

Libfabric AV Auth Key

Ian Ziemba

October 3, 2023

Overview

- Objective of PR 9319 is to support multiple authorization keys per endpoint
  - Previous proposal introduced new authorization key ring object
  - New proposal focuses on extending AV functionality
- Highlights of changes
  - Add fi_addr_t src_addr to struct fi_cq_err_entry
  - Add size_t max_ep_auth_key to struct fi_domain_attr
  - Add uint64_t optional_caps to struct fi_tx_attr, fi_rx_attr, fi_domain_attr, and fi_info
  - Add FI_AV_AUTH_KEY support
  - Add FI_AV_USER_ID as a primary capability

Add fi_addr_t src_addr to struct fi_cq_err_entry

- src_addr: Used to return source addressed related information for error events
  - How this field is used is error event specific
- FI_AV_AUTH_KEY uses this field to report auth key for FI_EADDRNOTAVAIL events
  - Discussed more later
- Providers need to check user defined API version to ensure ABI compat
  - Providers are encouraged to use ofi_cq_err_memcpy() to handle ABI compat

Add size_t max_ep_auth_key to struct fi_domain_attr

- max_ep_auth_key: Used by providers to report the number of authorization keys supported per connectionless endpoint
- If max_ep_auth_key > 1, FI_AV_AUTH_KEY can be used to support multiple auth keys per EP
  - If FI_AV_AUTH_KEY == 1, existing auth key interfaces can be used
- If providers support max_ep_auth_key > 0, they are required to implement FI_AV_AUTH_KEY

Add uint64_t optional_caps to fi_tx_attr, fi_rx_attr, fi_domain_attr, and fi_info

- optional_caps: Enables users to provide primary and secondary capabilities which are treated as optional
  - If requested, providers are not required to support the capability
  - If providers do not support the capability, the capability will be cleared in the corresponding fi_info caps fields returned from fi_getinfo
  - If providers do support the capability, providers will set the capability in the corresponding fi_info caps
- Motivation for this extension is to handle FI_AV_USER_ID not being identified as primary or secondary capability
  - Current precedence for how FI_AV_USER_ID operates is that it is just a flag passed into AV insert
  - What is missing is users do not know if FI_AV_USER_ID is supported by providers or not

Add FI_AV_AUTH_KEY support

- FI_AV_AUTH_KEY is set by libfabric users via fi_domain_attr::auth_key_size to denoted if MR and EP authorization keys come from the AV instead of MR and EP attrs
  - When set, providers will ignore fi_ep_attr::auth_key during endpoint enable
  - For MRs, fi_mr_regattr() must be used with fi_mr_attr::auth_key pointing to a struct fi_mr_auth_key and fi_mr_attr:auth_key_size equal to sizeof(struct fi_mr_auth_key)
    - fi_mr_auth_key::av should point to the AV the MR authorization keys should come from
    - If the domain is configured with FI_DIRECTED_RECV, fi_mr_auth_key::key is used to restrict the MR to a specific authorization key
- fi_av_insert_auth_key() is used to insert an auth key into the AV
  - Output is an fi_addr_t handle specific to this authorization key
  - If the EP is configured with FI_DIRECTED_RECV, this fi_addr_t can be used to match all EP addrs associated with this authorization key
  - Calling fi_av_remove() with this fi_addr_t will delete the authorization key
  - -FI_EBUSY will be returned from fi_av_remove() should this key still be used by an EP
- Once the AV is bound to an EP and the EP is successfully enabled, the EP will be configured to support all auth keys in the AV at that point in time

Add FI_AV_AUTH_KEY support cont

- Users must provide an authorization key fi_addr_t with fi_av_insert_{addr, svc, sym}
  - This is done by using the fi_addr arg as input
    - This conflicts with how FI_AV_USER_ID is defined today
  - The output of fi_av_insert_{addr, svc, sym} is an fi_addr_t mapping to a specific <EP addr, auth_key> tuple
- For FI_EADDRNOTAVAIL CQ errors, fi_cq_err_entry::src_addr will return the authorization key handle associated with the incoming data transfer
  - This, combined with the existing behavior of fi_cq_err_entry::err_data enables users to generate a fi_addr_t mapping to the specific <EP addr, auth_key> tuple which triggered the FI_EADDRNOTAVAIL event

Add FI_AV_USER_ID as a primary capability

- FI_AV_USER_ID: Indicates that the domain supports the ability to open address vectors with the FI_AV_USER_ID flag
  - If this domain capability is not set, address vectors cannot be opened with FI_AV_USER_ID
- For address vectors opened without FI_AV_USER_ID, user-specify fi_addr_t (e.g. fi_av_insert) through the fi_addr parameter.
  - The fi_addr parameter values are provided as part of address insertion (e.g. fi_av_insert) acts as input/output in this case (existing behavior)
  - Can lead to unexpected runtime failures
- For address vectors opened with FI_AV_USER_ID, by default, all inserted EP addrs will be assigned the user ID FI_ADDR_NOTAVAIL
  - fi_av_set_user_id() is used to defined the user-specified fi_addr_t
  - Specifying FI_AV_USER_ID per insert not supported
- If users require FI_AV_AUTH_KEY and FI_AV_USER_ID support, using the FI_AV_USER_ID flag per insert is not supported

Thank you

Email: ian.ziemba@hpe.com