



## **Appendix – Security Features**

# Table of contents

## SA Enhanced Trust Model (SAETM)

---

Set of Untrusted SA Requests Allowed

---

Proxy SA Requests

---

Registration Limits

---

SAETM Logging

---

## SGID Spoofing

---

## M\_Key Authentication

---

M\_Key Per Port

---

## SA Enhanced Trust Model (SAETM)

Standard SA has a concept of trust-based requests on the SA\_Key that is part of each SA MAD. A **trusted request** is when the SA\_Key value is not equal to zero but equals the SA configured value, while an **untrusted request** is when the SA\_Key value equals zero in the request. If a request has a non-zero SA\_Key value that is different from the configured SA key, it will be dropped and reported.

When SAETM is enabled, the SA limits the set of untrusted requests allowed. Untrusted requests that are not allowed according to SAETM will be silently dropped (for the set of untrusted requests allowed, see [the following section below](#)).

SAETM feature is disabled by default. To enable it, set the `sa_enhanced_trust_model` parameter to TRUE.

### *Additional SAETM Configuration Parameters*

Parameter	Description
<code>sa_etm_allow_untrusted_guidinfo_rec</code>	Defines whether to allow GUIDInfoRecord as part of the SAETM set of untrusted requests allowed (see <a href="#">section below</a> )
<code>sa_etm_allow_guidinfo_rec_by_vf</code>	Defines whether to drop GUIDInfoRecord from non-physical ports (see <a href="#">section below</a> )
<code>sa_etm_allow_untrusted_proxy_requests</code>	Defines the behavior for proxy requests (see <a href="#">section below</a> )
<code>sa_etm_max_num_mcs/ sa_etm_max_num_srvcs/ sa_etm_max_num_event_subs</code>	Defines the registration limits in SAETM (see <a href="#">section below</a> )

## Set of Untrusted SA Requests Allowed

The following table lists the untrusted requests allowed when SAETM is enabled:

Request	Request Type
MCMemberRecord	Get/Set/Delete
PathRecord	Get
PathRecord	GetTable (only if both destination and source are specified (e.g. only point to point))
ServiceRecord	Get/Set/Delete
ClassPortInfo	Get
InformInfo	Set (for non-SM security traps)
GUIDInfoRecord	Set/Delete – this request can only be part of this set depending on the values of sa_etm_allow_untrusted_guidinfo_rec and sa_etm_allow_guidinfo_rec_by_vf – see elaboration below.

When sa\_etm\_allow\_untrusted\_guidinfo\_rec is set to FALSE (and SAETM is enabled), the SA will drop GUIDInfoRecord Set/Delete untrusted requests.

When sa\_etm\_allow\_guidinfo\_rec\_by\_vf is set to FALSE (and SAETM is enabled), the SA will drop GUIDInfoRecord Set/Delete requests from non-physical ports.

If sa\_etm\_allow\_untrusted\_guidinfo\_rec=FALSE, GUIDInfoRecord Set/Delete requests will become part of the SAETM set of untrusted requests allowed. Note that if sa\_etm\_allow\_guidinfo\_rec\_by\_vf=FALSE, the requests will only be allowed from physical ports.

## Proxy SA Requests

SA modification request (SET/DELETE) is identified as a proxy operation when the port corresponding with the requester source address (SLID from LRH/SGID from GRH) is different than the port for which the request applies:

- For MCMemberRecord, when the MCMemberRecord.PortGID field does not match the requester address
- For ServiceRecord, when the ServiceRecord.ServiceGID field does not match requester address
- For the GUIDInfoRecord, when the LID field in the RID of the record does not match the requester address

When `sa_etm_allow_untrusted_proxy_requests` is set to `FALSE` and SAETM is enabled, untrusted proxy requests will be dropped.

## Registration Limits

When any of `sa_etm_max_num_mcgs`, `sa_etm_max_num_srvcs` or `sa_etm_max_num_event_subs` parameters is set to 0, the number of this parameter's registrations can be unlimited. When the parameter's value is different than 0, attempting to exceed the maximum number of registrations will result in the request being silently dropped. Consequently, the requester and request info will be logged, and an event will be generated for the Activity Manager.

The following parameters control the maximum number of registrations:

Parameter	Description
<code>sa_etm_max_num_mcgs</code>	Maximum number of multicast groups per port/vport that can be registered.
<code>sa_etm_max_num_srvcs</code>	Maximum number of service records per port/vport that can be registered.
<code>sa_etm_max_num_event_subs</code>	Maximum number of event subscriptions (InformInfo) per port/vport that can be registered.

## SAETM Logging

When requesting an operation that is not part of the SAETM set of untrusted requests, it will be silently dropped and eventually written to the SM log.

The logging of the dropped MADs is repressed to not overload the OpenSM log. If the request that needs to be dropped was received from the same requester many times consecutively, OpenSM logs it only if the request number is part of the following sequence:

0, 1, 2, 5, 10, 20, 50, 100, 200... (similar to the trap log repression).

## SGID Spoofing

SA can validate requester addresses by comparing the SLID and SGID of the incoming request. SA determines the requester port by the SLID and SGID field of the request. SGID spoofing is when the SGID and SLID do not match.

When `sa_check_sgid_spoofing` parameter is enabled, SA checks for SGID spoofing in every request that includes GRH, unless the SLID belongs to a router port in that same request. In case the request SGID does not match its SLID, the request will be dropped. The default value of this parameter is TRUE.

## M\_Key Authentication

### M\_Key Per Port

This feature increases protection on the fabric as a unique M\_Key is generated and set for each HCA, router, or switch port.

OpenSM calculates an M\_Key per port by performing a hash function on the port GUID of the device and the M\_Key configured in `opensm.conf`.

To enable M\_Key per port, set the parameter below in addition to the parameters listed in the [previous section](#):

```
m_key_per_port TRUE
```