SA Migration

IPsec Workshop 2026

Tobias Brunner

16.07.2026

Overview

- The Problem
- Previous Approaches
- The Solution

The Problem: IP Address Changes

- With MOBIKE, IP addresses of IKE/IPsec endpoints may change, especially clients can roam to a different IP address/family and out/behind NATs
- Does not require MOBIKE, IKEv2 has an implicit address update mechanism
- IP addresses of IPsec SAs **cannot** be updated via `XFRM_MSG_UPDSA`

strongSwan's Approach

1. Handle policies:
   - Install high-priority outbound drop policies for old TS
   - Remove the existing policies for all directions
   - Update the TS and possibly allocate a new reqid if they changed
2. Update the SAs:
   - Get the existing SA (including **keys**) and replay state (separate request)
   - Delete the SA
   - Install SA again with new addresses, selector, encap state, reqid, etc., while preserving keys and replay state
3. Update policies:
   - Install policies for the new TS
   - Remove the drop policies for the old TS

Issues With strongSwan's Approach

- Migrating a Child SA requires **eight** messages to query (SAs and replay states), delete, and re-install
- Doesn't work if kernel's *lockdown* feature is set to **confidentiality**
  - Keys in queried SA are zeroed out $\rightarrow$ newly installed SAs broken
  - Will be handled more gracefully with next update (fallback to rekeying)

The XFRM_MSG_MIGRATE Message

- `XFRM_MSG_MIGRATE` provides a way to change the IP addresses of SAs and policies and their templates in one go
- Was originally intended for Mobile IPv6 (MIPv6 daemon manages policies, IKE daemon SAs):
  - Lookup requested policy (exit if not found)
  - Find matching SA(s) and migrate each (fine if none found)
  - Migrate policy templates only
  - Delete old SAs

Issues with XFRM_MSG_MIGRATE: Lookup

- Entry is via policy, which might be managed by the user or a third-party daemon
- Policies are shared by multiple SAs in some scenarios (e.g. per-CPU SAs or SELinux labels where a generic policy directs traffic to multiple SAs that are selected based on CPU ID or specific label)
- On the other hand, with IKEv2, there could be a lot more policies than SAs, as each combination of local/remote TS requires two to four
- SA lookup is based on addresses, protocol, mode, and optionally reqid, which makes targeting specific SAs impossible

Issues with XFRM_MSG_MIGRATE: Migration

- Policy selectors can't be changed (host-to-host/net tunnels), neither can SA selectors
- Impossible to remove/disable UDP-encapsulation
- Changing reqids is not supported
- Race condition (old SA may still get used after it got cloned) → duplicate seq/IV

The Solution: XFRM_MSG_MIGRATE_STATE

- Thanks to Antony's work over the past two years, we now have a solution:

  `XFRM_MSG_MIGRATE_STATE`
- **Migrates a single, specific SA** (lookup via SPI, destination address, protocol, and mark)
- As usual, takes a struct (`xfrm_user_migrate_state`) and several optional attributes
- Properties of the old SA are generally preserved if no new value is passed
- Flags passed in the struct control some exceptions to that
- Available with Linux v7.2
- Antony also fixed issues with `XFRM_MSG_MIGRATE`, e.g., encap removal, address family change, race condition

Things XFRM_MSG_MIGRATE_STATE Can Migrate

- Passed via struct (required, not inherited):
  - IP Addresses
  - SA selector (`XFRM_MIGRATE_STATE_UPDATE_H2H_SEL` flag inherits previous selector and updates addresses)
  - Reqid (0 means no reqid)
- Passed via attribute (omit to inherit):
  - Mark (`XFRMA_MARK`)
  - Output mark (`XFRMA_SET_MARK`)
  - UDP encapsulation (`XFRMA_ENCAP`, `encap_type=0` to remove/disable)
  - NAT-T keepalive interval (`XFRMA_NAT_KEEPALIVE_INTERVAL`, only if encap enabled)
  - Mapping change events threshold (`XFRMA_MTIMER_THRESH`, only for inbound SAs)
  - Hardware offload device (`XFRMA_OFFLOAD_DEV`, `XFRM_MIGRATE_STATE_CLEAR_OFFLOAD` flag to disable offloading)

Changes in strongSwan

- `XFRM_MSG_MIGRATE_STATE` is used on v7.2+ kernels (hard check)
- Updating SAs now requires **two** instead of **eight** messages (also smaller)
- The rest of the algorithm remains unchanged for now, but due to the atomic update, we might change policy handling in the future
- On older kernels, fallback to rekeying if *lockdown*=**confidentiality**

References I

- strongSwan.
  https://strongswan.org.
- XFRM_MSG_MIGRATE_STATE Documentation.
  https://www.kernel.org/doc/Documentation/networking/xfrm/xfrm_migrate_state.rst.