I2NSF & IPsec interm

Use case 2 issues

Tero Kivinen
kivinen@iki.fi

Implementation issues

- Implementing IKE is actually simple, implementing the policy code required to run IKE and tie it to the IPsec is more complicated than actual KMP itself
  - In use case 2 the security controller needs to implement this policy code regardless whether it implements IKE
  - It cannot reuse existing implementations as current code is usually very tightly integrated to existing IKE and IPsec

Chicken and egg problem

- As the keys are now generated in the Security controller, they need to be transferred to the NSF in secure manner, which would require thing like IPsec to protect them…, but we cannot use IPsec as we are transferring IPsec keys.
- If TLS or similar is used to transfer IPsec keys, then the device is no longer constrained, as TLS KMP is about same level than IPsec.

NSA key storage

- If nation-state adversary (NSA for short) wants to get all the keys used by network the security controller offers NSA perfect target for attack, or perfect place to require hooks for key storage i.e., we are putting all the eggs in same basket.
- Security controller will most likely store the configuration (including keys) to the stable storage (so if security controller reboots it still knows how to reconfigure network), and that stable storage will be backed up, as those are mandatory requirements for high availability systems.
- Those stable storage hard disks, backup tapes, or cloud backup servers are perfect ways to get the traffic keys for the whole network.
- Actually as no cryptographic attacks are required this is not only for NSAs but all other attackers also.

Nonce reuse

- Modern ciphers (AES-CGM, ChaCha20 etc) require that nonce is never reused, and MUST be used with automated key management, and they forbid using statically configured keys.
  - From the NSF point of view it IS using statically configured keys, as it has no way of knowing how security controller generated keys
  - Also when NSF restarts it MUST NOT reuse the existing keys it might have from before restart, but needs to get fresh keys from the security controller. Security controller CANNOT just send previously generated data to NSF, it MUST generate fresh keys and distribute them to all peers.

Latency issues

- There are some triggers or events generated by the IPsec which are time sensitive, i.e., they need to be processed without extra delay.
  - For example creating new SA because trigger rule was hit
  - Or rekeying SA because it is reaching its byte count limit
- Requiring NSF ↔ security controller interaction for each of those will cause issues