3GPP TSG SA WG3 Security — S3#21 S3-010643

27 - 30 November, 2001

Sophia Antipolis, France

Source: Joint Alcatel/Siemens

Title: Use of Push vs Pull Mechanisms in local SA distribution

Document for: Decision

Agenda item: T.b.d.

1 Introduction

During meeting S3#20, document S3-010507 argued for the necessity to adopt a mixed push/pull approach for the distribution of policy information and SA information towards the NEs within a domain. The Draft Report of the Sidney Meeting (S3-010563) says: “The issue was considered too involved to make a decision at this meeting, although the availability of both push and pull mechanisms in both databases seemed useful. Further analysis should be done before making decisions on this. It was considered that the usual case for both databases should be push, and in exception cases pull. **It was decided that an e-mail discussion should be made to try to agree on a mechanism, and contribution, on detailed mechanisms and analysis of them, should be made to the next meeting.”**

This contribution further elaborates on the pros and cons of each possible approach to show that the best solution is to adopt a default push mechanism, supplemented by extensions for exceptional cases.

2 Pure Pull Mechanism

In this approach, originally documented in TR 33.800, the NE is solely responsible to determine that it needs SA information that it does not currently hold. We further examine the Pull mechanism for each situation under which a NE may need to get MAPsec SA information from the KAC.

1) The NE needs to apply a MAPsec SA for some outgoing MAP message and determines that this MAPsec SA has expired (soft-expiry or hard-expiry time is passed). The NE sends a Pull-request to its KAC to retrieve a valid MAPsec SA replacing the expired one. This policy brings a major disadvantage in that it implies a latency in the processing of the outgoing MAP message, waiting for the new MAPsec SA. Depending on whether the KAC already holds the new MAPsec SA or it needs to set it up with the peer domain KAC, latency will be longer.

2) The NE determines that a MAPsec SA has expired as soon as it actually expires (via some background task). The NE sends a Pull-request to its KAC to retrieve a valid MAPsec SA replacing the expired one. A major disadvantage of this approach is that the KAC may get overloaded with requests from all NEs coming in at the same time.

3) The NE receives a MAPsec message for which it does not hold the MAPsec SA. Current procedures in annex A of TS 33.200 require that the receiving NE finds a VALID MAPsec SA as pointed by the SPI in the received MAPsec message. Otherwise, the MAPsec message is rejected. This requires that the NE must be aware of any new inbound MAPsec SA as soon as one has been set up and can potentially be used by a peer entity. However, the current procedures do not foresee the case where the receiving NE would try and retrieve this unknown SA from the KAC. Case 2 above solves this issue as the NE would automatically retrieve a new SA when the current one expires but still it does not solve the problem for brand new SAs which are not follow-ups of existing ones.

4) An existing MAPsec SA needs to be “cancelled” prematurely (such as because it is compromised). A pure Pull approach does not make it possible for the KAC to distribute a new SA replacing (ie cancelling) the existing one.

5) In addition to the above considerations related only to the distribution of MAPsec SAs, it is important to also consider that policy information (SPD) must be made available to NEs since this is the basis for decisions taken by NEs when processing MAP messages. There is no way for a NE to determine when to retrieve any update to its local SPD. Such policy information must therefore necessarily be pushed by the KAC when an update has occurred in the policy (possibly together with new associated MAPsec SAs). Or at least the KAC must warn the NE of such an update to the policy, which can then be retrieved by the NE. However, the latter scenario unnecessarily adds an extra message to the distribution mechanism.

In conclusion, the above analysis shows that a pure Pull approach does not stand as it is not sufficient to cover all situations.

3 Pure Push Mechanism

In this approach, the KAC is solely responsible to determine that it needs to distribute new SA information towards NEs. We further examine the Push mechanism for each situation under which the KAC needs to distribute an update.

1) The KAC detects that an existing outbound MAPsec SA is reaching its expiration time and a new MAPsec SA is therefore set up with the corresponding peer domain KAC. At some point in time, the KAC pushes the new MAPsec SA towards all NEs (at least those which have relationships with the peer domain). The KAC ensures that the distribution occurs before the existing MAPsec SA actually expires in the NEs. This ensures service continuity at the NEs as the NE always has a MAPsec SA to use for outgoing MAP messages.

2) A new inbound MAPsec SA has been created to replace an existing one which is reaching its expiry time. At some point in time, the KAC pushes the new MAPsec SA towards all NEs (at least those which have relationships with the peer domain). The KAC ensures that the distribution occurs before the existing inbound MAPsec SA actually expires in the NEs. This ensures service continuity at the NEs. This is especially important as current procedures would make the NE to reject incoming MAP messages protected under the new SA.

3) A new outbound MAPsec SA has been created to replace and cancel an existing SA before it reaches its expiry time. The KAC immediately pushes the new SA (also identifying the cancelled SA) to the concerned NEs. This ensures that the new outbound SA can be used immediately by the NEs.

4) A new inbound MAPsec SA has been created to replace and cancel an existing SA before it reaches its expiry time (such as because it is compromised). The KAC immediately pushes the new SA (also identifying the cancelled SA) to the concerned NEs. It may be that the peer KAC was faster in distributing the new SA in its own domain and hence a peer NE used the new SA before the NE in the local domain knows about it. Current procedures make the receiving NE to reject the MAPsec messages as long as it does not learn about the new SA. This behavior is inherent to revocation, not to the push mechanism. The pure push mechanism provides a method for revocation, the pure pull mechanism does not.

5) When the policy information (ie SPD) is updated, the KAC pushes the new information towards the NEs. This can be together with updated information on SAD (new or deleted SAs).

If we now take the NE’s point of view, we can consider the following cases.

6) The NE needs to apply a MAPsec SA for some outgoing MAP message and determines that this MAPsec SA has expired. A new SA will have been made available by the KAC before, as described in bullet 1 above.

7) The NE receives a MAPsec message for which it does not hold the MAPsec SA. As described in bullets 2 and 4 above, such a situation does not occur, except as described in bullet 4.

In conclusion, the above analysis shows that a pure Push approach looks more natural than a pure Pull approach and covers more situations (actually all but the exceptional case described in bullet 4 above).

4 Extended Push Mechanism

From our analysis in sections 2 and 3, it comes out that an Extended Push approach provides a complete solution.

Because the management (creation, cancellation, …) of policy information and MAPsec SAs is under control of the KAC, it is more efficient to adopt a default behaviour of Push (ie as described in section 3). This indeed avoids any possible delays in NEs. The NE could also be able to trigger the push of the unavailable MAPsec SA for outgoing messages to cover uncommon situations where it would not have received a MAPsec SA applicable to outgoing traffic towards a given peer domain.

5 Conclusion

As discussed above, the pure Pull approach is not sufficient, as it does not simply cover all situations. Morever, there is anyway a need for a Push mechanism to distribute policy information from the KAC towards the NEs. It is therefore natural to take advantage of that mechanism to also distribute SA information from the KAC when such SA information becomes available at the KAC (or when the KAC decides to do so). Because the KAC is the central node in this architecture, the default action should be for the KAC to push any updates to SAD and SPD when it becomes available (according to the operator's policy).

We therefore suggest that SA3 adopts an extended push approach for the intra-domain policy and SA distribution mechanism, as described in section 4.