3GPP SA2/SA3 joint meeting **Tdoc S3-010083**
Göteborg, Sweden
27th February 2001

Source: Ericsson

Agenda Item: TBD

Title: Providing the S-CSCF name to the P-CSCF

Document for: Discussion

---

1. Background

One of the roles of the Interrogating CSCF in the IP-MM domain is to act as the point of contact for in-bound/out-bound multimedia sessions destined for a user/subscriber of the operator owning that I-CSCF. This role includes an optional task of the I-CSCF to act as a proxy hiding the internal network architecture from the P-CSCFs of a visited network for competition and security reasons.
Since the I-CSCF is supposed to be stateless it would then have to inquire the HSS about the identity/address of the appropriate S-CSCF for each received in-bound SIP message.

The basic idea of both proposals (please refer to contributions S2-010325 and N1-010253) discussed in this contribution is to avoid loading the HSS by sending the identity/address of the S-CSCF to the P-CSCF. The P-CSCF would store this information and then include it in each subsequent SIP message forwarded from the user/subscriber in question to the I-CSCF.

2. Analysis of the “encryption” mechanism (S2-010325)

The S2-010325 contribution suggests that the identity/address of the S-CSCF is encrypted by the I-CSCF and forwarded to the P-CSCF at a positive user registration response. The key(s) used to encrypt and decrypt the identity/address would always be kept in the I-CSCF and never revealed to any entity outside the HN.

Since the purpose of the scheme is to hide information such as the number of S-CSCFs employed by the HN from the VN, it is important to add some sort of random parameter to the S-CSCF identity/address before encryption. If this is not done, the same S-CSCF identity/address would always yield the same cryptographic output from the I-CSCF, and the P-CSCF could by just comparing the number of different strings received from an I-CSCF estimate e.g. the number of S-CSCFs employed in that network.

In order to maintain the security level the encryption key(s) also need to be exchanged regularly. Since the encrypted data is sent to the P-CSCF at SIP registration and users will be registering at different points in time, it means that the I-CSCF entity has to be able to handle at least two different (sets of) keys simultanously. The number of keys necessary to support in an I-CSCF will depend on the relation between the re-registration timer and the chosen key exchange rate.

Thus, in order to avoid having the I-CSCF trying all possible valid keys, the data sent to and stored in the P-CSCF must also include a ‘key identity’ parameter identifying the key that was used encrypting the S-CSCF identity/address.

A method to avoid the problem of multiple keys is to define a mechanism with which the I-CSCF will be able to push an updated version of the encrypted data to all relevant P-CSCFs. Such a scheme would probably be very complex though, and require states or user information to be kept by the I-CSCF, which was not desired from the beginning.

According to the TS 23.228 there might be several I-CSCFs.
If the encryption/decrytion keys are generated by each I-CSCF itself no key distribution scheme is required, but in that case the identity of the I-CSCF that encrypted the data has to be forwarded to the P-CSCF as well, since it will be the only I-CSCF able to decrypt the data again. This will also cause the I-CSCF to become a single point of failure.
Another way is to let all the I-CSCFs of a network share the same encryption/decryption keys. This will require a quite complex key management mechanism though, and in itself pose the requirement on all I-CSCFs to be able to handle at least two keys during the period from the time when the key exchange procedure has begun, till the time when all I-CSCFs have been updated.

3. Analysis of the “token” mechanism (N1-010253)

The N1-010253 contribution proposes a more generic mechanism where the S-CSCF identity/address is passed to the P-CSCF in the form of a token. Since the information carried by the token is relevant only within the HN or I-CSCF, it can be created in any way. It could be the encrypted S-CSCF identity/address straight off as suggested in the S2-010325 contribution (please refer to the comments above), but it could also be just a serial number whose association with the address/identity of the S-CSCF can be found in some database of the HN (obviously not the HSS, though, since that would defeat the its original purpose).

The token could itself contain the identity of the I-CSCF that created the token (e.g. the n^{th} octet of the token could be the I-CSCF identity). This would enable the HN to employ multiple I-CSCFs without posing the requirement to distribute all “tokens” to all I-CSCFs, or to employ a centralized database containing all the currrently valid tokens.

Just as with the encryption mechanism, though, special consideration should be given to the design of the token so that tokens refering to the same S-CSCF will seem random to the P-CSCF, thereby ensuring that the architecture of the HN will not be revealed.

From a security point of view two different threats have been identified.

The token could be altered or removed during its storage in the P-CSCF. This would mean that the I-CSCF either gets a token it does not understand, or it does not get any token at all. Both this cases could probably be easily solved forcing re-registration from the HN by responding with a 407 (unauthorized access) message towards the terminal. When the terminal registers again a valid token will be created and sent to the P-CSCF. This procedure would then be exactly the same as the registration procedure for the case when the IMS AKA authentication mechanism is used.
It could of course be used as an indirect way to launch a Denial of Service attack towards the home network, but it is also true that there will be a lot of other, just as effective (and just as impossible to prevent), ways to achieve this anyway (e.g. by deleting packets during transit between the two networks).

The second threat is that the token gets stolen (i.e. copied) by an external party. But, as long as the token does not contain any value other than pointing out the appropriate S-CSCF in the HN, Ericsson have not be able to find any sort of risk connected to such a theft.

4. Conclusion

Ericsson have found that the “encryption” mechanism (described in S2-010325) imposes quite complex problems that need to be solved requiring a high degree of standardization efforts, while the “token” mechanism (described in N1-010253) is simpler to its character and does not require much further standardization efforts. Furthermore, it still allows the “encryption” solution to be used as one way to implement the token.

Ericsson believe that it from a security point of view generally is a bad idea to send data, that is supposed to be hidden, to the entity it should be hidden from. Therefore we propose not to introduce any additional mechanism (please refer to Ericsson’s contribution S2-010xyz).