3GPP TSG SA WG3 Security — S3#16 **S3-000693**

28-30 November, 2000

Sophia Antipolis, France

Source: **Vodafone**

Title: **MExE security issues**

Document for: **Decision**

Agenda Item: **9.3**

1 Introduction

There are a number of initiatives within MExE to broaden its scope by allowing MExE executables to have greater functionality and flexibility. There are clear advantages associated with these proposals. However, some of the proposals also raise some security concerns. This paper identifies the key security concerns and recommends solutions. It is proposed that these recommendations be considered for approval by S3 and that an appropriate liaison statement is sent to T2 MExE.

2 Overview of MExE security

A MExE device may download a MExE executable that is intended for either one of the three *trusted* secure execution domains (operator, manufacturer or third party) or for the *untrusted* domain. If the executable is targeted to one of the trusted domains, then that executable must be digitally signed so that it can be verified on the terminal using a trusted root public key corresponding to that domain via an appropriate certificate chain. Capabilities within each of the four domains are restricted according to a standardised list of permitted APIs.

3 The need to check the APIs in a MExE executable before runtime

The current specifications do not require a MExE terminal to check an executable before runtime to ensure that it does not contain any APIs which are not permitted in the executable's domain. If this checking is not done on the terminal and the executable contains APIs which are not permitted in the executable's domain, then there are two undesirable consequences:

- The executable may not execute properly because it tries to access APIs that are not permitted in its domain (e.g. a runtime error may occur with unpredictable effects)
- The risk is increased that a malicious executable can be written that can successfully exploit an implementation weakness in the terminal which allows an otherwise restricted API in the executable's domain to be used.

These problems are severe in the case of executables that are assigned to the untrusted domain since the executable's source cannot be reliably identified. This problem is further exacerbated in the case of untrusted pushed executables. Current models for pushed executables are based on notifying the user that an executable is awaiting download from a MExE server. The MExE server is not required to send any more information to the user regarding the pushed executable. Therefore the user will have very limited information on which to make a decision on whether to download and run the executable. A malicious executable developer could therefore easily use MExE push to propagate executables which do not execute properly or executables which exploit an implementation weakness on a device. The recent high profile email virus problems on the Internet have highlighted that many users may be fooled into running malicious executables which exploit weaknesses in software.

To solve these problems, it is proposed that a mechanism is added to the specifications to allow the MExE terminal to check an executable before runtime to ensure that it does not contain any APIs which are not permitted in the executable's domain.

Note that it is not possible for specifications to be written where this checking is done in the MExE server or in a network gateway, since the scope of the MExE specifications is restricted to the terminal itself.

4 The need to for proper procedures for assigning an executable to a particular domain

4.1 Assigning 'signed' executables to the untrusted domain

A proposal is currently under consideration in T2 MExE whereby executables that are targeted for the untrusted domain may be signed so that a terminal with a trusted root public key can verify the signature via an appropriate certificate chain. Furthermore it is proposed that when such executables are downloaded to terminals which do not support signature verification (e.g. Classmark 3 devices), they are still permitted to execute in the untrusted domain.

It is believed that this proposal abuses the definition of the untrusted domain and may lead to confusion by users. For instance, if a MExE Classmark 3 terminal downloads a signed executable targeted for the untrusted domain, then the presence of a signature may lead to misinterpretation by the user.

As a solution to this problem it is recommended that if executable developers want to be able to sign executables which are targetted for a domain which does not require the capabilities (APIs) of the trusted domains, then either one of the following solutions is adopted:

- the executable is signed so that it can be verified in one of the existing trusted domain (e.g. third party)
- the executable is not signed

If neither of these solutions is acceptable then it is suggested that a new trusted domain, which contains the same restrictions on APIs as the untrusted domain, could be considered for standardisation.

4.2 Assigning "trusted" executables with invalid signatures to the untrusted domain

Another proposal under consideration in T2 MExE is to allow executables targeted for trusted domains, whose signatures cannot be verified, to be able to execute in the untrusted domain.

The signature verification could fail because a trusted root key for the particular certificate chain might not be available on the terminal or because the executable was modified after the signature was applied.

A result of assigning the executable to the untrusted domain will be that the executable will only have limited functionality, as access to trusted APIs will be denied. If the executable contains APIs which are not permitted in the executables domain, then the two undesirable consequences listed in section 1 are applicable.

Rather than rely on a mechanism on the terminal to check the executable's APIs before runtime, it is recommended that an alternative solution is adopted whereby the executable is simply deleted if the signature verification fails. This option is recommended since it keeps the MExE security model simple, clear and consistent. With this solution the procedure for verifying a MExE executable is as shown in Figure 1.

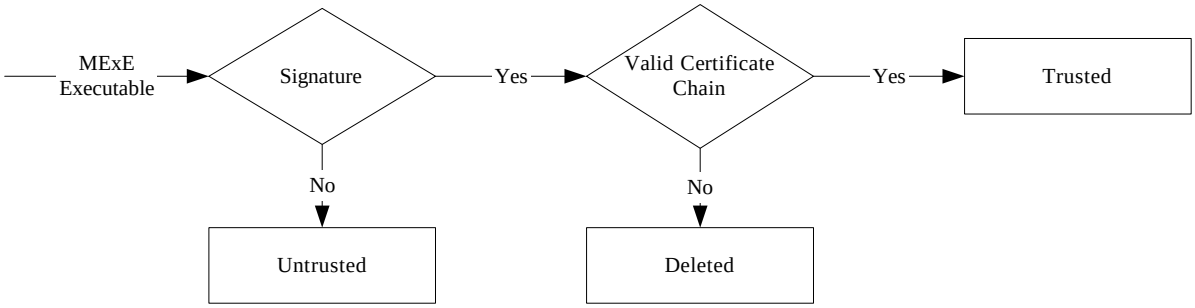

Figure 1: Recommended procedure for verifying a MExE executable

5 Conclusion

It is proposed that the following recommendations are approved by S3 and that an appropriate liaison statement is sent to T2 MExE:

- A mechanism is added to the specifications to allow the MExE terminal to check an executable before runtime to ensure that it does not contain any APIs which are not permitted in the executable's domain.
- If executable developers want to be able to sign executables which are targetted for a domain which does not require the capabilities (APIs) of the trusted domains, then either one of the following solutions is adopted:
    - the executable is signed so that it can be verified in one of the existing trusted domain (e.g. third party).
    - the executable is not signed.

  If neither of these solutions is acceptable then it is suggested that a new trusted domain, which contains the same restrictions on APIs as the untrusted domain, could be considered for standardisation.
- An executable is deleted if the signature verification fails rather than being assigned to the untrusted domain.