3GPP TSG WG SA3 –Security SA3#16 **S3-000770**
28-30 November, 2000
Sophia Antipolis, France

Title: **MExE security issues**

Source: **TSG SA WG3**

To: **TSG T WG2 MExE**

Contact Person:

Name: Colin Blanchard, MExE security work item rapporteur, TSG SA WG3

colin.blanchard@bt.com

S3 would like to thank T2 MExE for their LS (S3-000704=T2-000738) which contained comments on the security concerns highlighted in a Vodafone contribution to S3#16 (S3-000693=T2-000735).

S3 offer the following response:

- On the concept of a pre-runtime check on whether an executable makes use of functions which are restricted in a particular domain, S3 acknowledge T2 MExE's comments on the difficultly of implementing such a security mechanism on terminals.
- S3 have noted the possibility of allowing 'signed' executables to run in the untrusted domain when the algorithmic signature check is successful but the public signature verification key cannot be verified back to a root public key of a secure domain. S3 have not yet studied the security aspects of this proposal..

S3 welcome continued co-operation with T2 MExE. As part of the S3 work item on MExE security, S3 would like to invite T2 MExE to participate in a forthcoming S3 meeting with the specific objective of ensuring that security issues continue to be addressed as new releases are developed (in particular Release 4).

S3 meeting schedule:

Meeting	Date	Location	Host
*S3#17	27 February - 1 March 2001	Sophia Antipolis, France	ETSI Secretariat
S3#18	21 or 22 – 24 May 2001	Phoenix, Arizona (TBC)	Motorola (TBC)
S3#19	3 or 4 - 6 July 2001	London (TBC)	Vodafone (TBC)
S3#20	15 or 16 – 18 October 2001	Madrid (TBC)	Ericsson (TBC)

* The date, location and host of this meeting may change to facilitate a joint meeting with S2.

Attachments: S3-000704 (with S3-000693=T2-000735 enclosed)

3GPP TSG SA WG3 Security — S3#16 **S3-000704**

28-30 November, 2000

Sophia Antipolis, France

3GPP TSG-T2 #11 ***T2-000738***

Shin Yokohama, Japan, 27th Nov-1st Dec 2000

Liaison Statement

From: TSG T2 MExE

To: TSG S3

Subject: MExE group comments to "MExE security issues" Vodafone document

Contact: Mark Cataldo (mcatald1@email.mot.com), +44 777 5582288

---

1 Introduction

The MExE group has been informed (just one working day before the start of its meeting) of the Vodafone intention to submit a document to S3 (attached below). MExE was regrettably not given the opportunity to address the contents of the attached document to S3. *This* MExE LS gives a broader description of the ongoing work in T2 MExE, and directly addresses the content of the attached document to S3.

The MExE group apologises for the length of this LS, however of greatest concern to MExE is that the attached document to S3 is misleading and unrepresentative of the actual facts. The MExE group requests that the contents of *this* LS are carefully considered. The MExE group would have liked to present this LS to S3, but due to a concurrent MExE meeting in Japan, and the extremely short notice given, is logistically unable to attend the S3 meeting.

2 MExE concerns with the attached document

Having reviewed the attached document to S3, the MExE group concludes that it contains a confusing mixture of:-

- private e-mail discussions not pertinent to the MExE specification
- misunderstanding and misconceptions of current MExE work and discussions
- discussion of Release 5 issues not proposed for Release 4
- proposed (i.e. unagreed) CRs which are in discussion and not part of the MExE specification

Further, the attached document to S3 proposes major structurual changes which not only would fundamentally affect Release 98, Release 99 and Release 4, but which

- have not been formally proposed in MExE meetings,
- contradict the current architecture, and
- contradict the S3 support of the same MExE architecture for Release 98 and Release 99

The integrity and consistency of the information provided in the attached document to S3 represents a distorted view of the MExE Release 4 specification (as defined by the set of internally agreed CRs). The attached document to S3 is therefore inconsistent with the status of the MExE specification itself, contains errors in its contents, and could potentially misinform S3. The MExE group has therefore agreed to immediately send *this* LS to avoid any misunderstandings within S3.

3 Response to points raised

There are several issues which are either misrepresentative or do not appear to have been fully understand in the attached document to S3, and these are directly addressed below.

- *Section 2 Overview of MExE security of the attached document, states that "… Capabilities within each of the four domains are restricted according to a standardised list of permitted APIs …".*

  This statement is not correct. The principle of the MExE security domains is that once the authenticity and integrity of a signed executable has been verified (by verifying the digital signature with the public key on the MExE device), it is then assigned to one of the secure domains (by verifying a certificate

chain from the public key that was used to verify the signature back to a root public key which controls one of the security domains).

There is no standardised list of permitted APIs that the executables are permitted to use. Instead MExE explicitly identifies the sensitive functionality which shall not be accessed by the executables. This approach enables a far more secure targeted support of restrictive executable functionality.

This principle of

- identifying which functionality shall not be accessed by executables
- identifying which functionality is permitted for which secure domain (e.g. operator domain has wider range of functional access)

is continued from Release 98 and 99, which S3 reviewed and fully supported.

- *Section 3 The need to check the APIs in a MExE executable before runtime of the attached document, states "… 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 …"*

  The MExE specification does not take this approach, and thus nothing is new or changed; indeed the MExE approach of identifying which functionality shall not permitted in the security domains continues from Release 98 and 99, which S3 reviewed and fully supported.

  The proposed approach (which has never been formally presented to MExE, was partially discussed via e-mail between a few individuals, and not in any MExE meeting or on the MExE reflector), is incompatible with the architecture of small mobile devices, and would add significant load and latency. The proposed approach could be theoretically feasible, however it could not be absolutely certain to capture 100% of instances, and it would still require runtime Runtime API verification to be done!

  Section 3 of the attached document goes on to state "… *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)*…". This is quite a legitimate way for executables to be controlled and restrained; the objective of MExE is to not permit access to controlled functionality rather than static code checking in advance.

  These executables are signed executables running in secure domains, and the parties which generated these executables have had them signed by a trusted owner of one of the secure domains (e.g. an operator). The implicit statement that this approach is insecure is not correct.

- *Section 3 The need to check the APIs in a MExE executable before runtime of the attached document, states "… 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…"*

  Nothing is new or changed here; indeed this general approach continues from Release 98 and 99, which S3 reviewed and fully supported.

  The MExE requirement is that executables operating in these secure domains shall not be permitted access to sensitive functionality. It is a compliance requirement that this is fulfilled, and thus MExE devices would not permit the kind of exposure which has been suggested.

- *Section 3 The need to check the APIs in a MExE executable before runtime of the attached document, state "… 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 …"*

  Nothing is new or changed here; indeed this general approach continues from Release 98 and 99, which S3 reviewed and fully supported.

  MExE has a severely restricted set of functionality for untrusted executables. The approach taken is that the executable is an unknown entity, and regardless in what form this executable arrived on the device, running in the untrusted domain means it is highly controlled and restricted.

- *Section 3 The need to check the APIs in a MExE executable before runtime of the attached document, states "… 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 …"*

  This general approach for the untrusted domain continues from Release 98 and 99, which S3 reviewed and fully supported.

  As described above, when an executable is running in the untrusted domains, it is exposed to extremely limited functionality, and it is not permitted access to any sensitive functionality.

On the issue of pushed executables, the attached document addresses pushed MExE executables which are currently under investigation within MExE, are not covered in any aspect and will be studied in more detail as part of Release 5. This will form part of the push activities taking place in the WAP Forum and in 3GPP and thus preliminary discussion.

- *Section 4.1 Assigning 'signed' executables to the untrusted domain of the attached document, appears to misunderstand the technical debate currently taking place in MExE.*

  The issue concerns an executable which is signed, undergoes a successful algorithmic signature verification, but unsuccessfully attempts to be assigned to a secure domain (because the public key used to check the signature of the executable does not have a complete key chain back to a root public key controlling access to a secure domain).

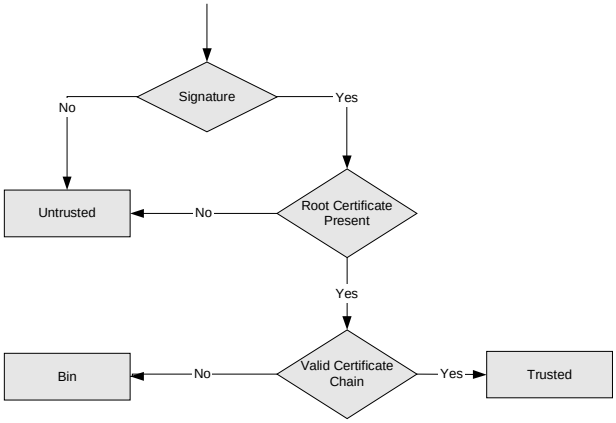
  In this unique example, MExE is currently considering the option of potentially allowing this executable to be permitted to run in the severely restricted untrusted domain, potentially after interrogating the user. If the user rejects this proposal, then the executable would be discarded, whereas if the user accepts this proposal then the executable would be treated just as any other untrusted executable (i.e. tightly controlled, severely limited functionality access etc.). In other words, the executable would be treated as if it had never been signed at all.

  There would be no confusion in the user's mind as to its status, as the user would be directly queried as to whether he wishes to have it as an untrusted executable. Further, running in the untrusted domain, this executable has severely limited access to functionality, despite the fact that its signature (see above) had been successfully verified.

  MExE is not only considering this proposal following an initial Vodafone proposal, but also because service providers may wish to have simple executables which could potentially run both in a secure and untrusted domains, but the service provider may not have maintained both signed and unsigned copies of the executable in their servers (duplication!). Further, with the introduction of Classmark 3, service providers may wish to have a simple executable which would execute both in a secure operator domain as well as in a untrusted domain, without having to have the very same executable both with and without a digital signature in the server.

  In fact, the arguments used in the attached document *against* this discussion topic are very confused, because it was Vodafone that first proposed this approach (the following figure and text is a direct extract from T2x00102 (from MExE's September 2000 Vuokatti meeting).

The diagram below (*Diagram 2*) is proposed as an alternative to *diagram 1*. It should be pointed out that integrity checking is also achieved in this proposal. This is achieved when the certificate chain is verified. This diagram clearly shows that failure in certificate chain implies that the application is untrusted and also it that could have been tampered. Another difference in this proposal is that it checks for the root certificate before it checks for the certificate chain. This will clearly differentiate between the two scenarios where certificate chain fails due to the lack of a root certificate and certificates having been tampered with.



  Debate on this issue is still ongoing in MExE, and no decision has been taken. MExE is probably one of the most security conscious groups outside S3, and will carefully examination the issues involved. Indeed, security experts (including from S3) regularly attend MExE meetings.

- *Section 4.2 Assigning "trusted" executables with invalid signatures to the untrusted domain in the attached document*

  The attached document appears to misunderstand the technical debate currently taking place in MExE. MExE is not considering such a proposal, and it would appear that the attached document has misunderstood considerations of the issue as described in section 4.1 of the attached document and as described above.

4 Conclusion

The MExE group is continuing its exhaustive efforts to provide a very secure framework for terminal executables. The MExE group concludes the following:-

1. The comments in the attached document would appear to raise concern. However when read in context and bearing in mind that security in the MExE specification basically remains unchanged from Release 99, it is clear that it is a case of misunderstanding the issues and current discussion topics rather than a change in MExE security. Thus, the attached document to S3 is technically incorrect in many of its presumptions.
2. The attached document is proposing significant MExE security changes which have already been reviewed, accepted and agreed by S3 for MExE Release 98 and 99. The MExE group considers this an unreasonable approach given the S3 support of MExE in Release 98 and 99.
3. A new area currently being discussed is the item (referred to in section 4.1 of the attached document to S3 and answered above) is "signed" executables running in the untrusted domain. This is the unique case of an executable which had a successful algorithmic signature check, but whose public key cannot be verified back to a root public key of a secure domain. Note that had this very same executable been sent to the MExE device without a signature, then it would be permitted to run in the untrusted domain anyway! These proposals are not agreed and not part of the MExE specification. MExE requests, and would welcome, feedback from S3 on this issue prior to making a decision, and in the interim will send further details on this matter to S3.

The MExE group assures S3 of its continued diligence regarding security, and looks forward to its continued co-operation with S3.

"T2-000735
(Vodafone document

3GPP TSG-T2 #11 ***T2-000735***

Shin Yokohama, Japan, 27th Nov-1st Dec 2000

3GPP TSG SA WG3 Security — S3#16 **S3-00XXXX**

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.

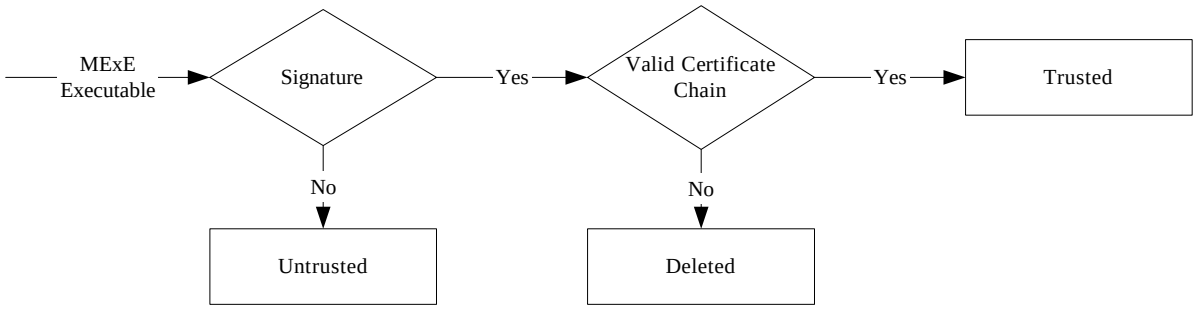


Figure 1: Recommended procedure for verifying a MExE executable

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.