

**Method for controlling an embedded calculator adapted to control a critical system
and associated calculator and vehicle**

5 This patent application claims the benefit of document FR 22 14531 filed on December 17, 2022, which is hereby incorporated by reference.

 The present invention relates to an embedded calculator and to a vehicle comprising such an embedded calculator.

 In the field of embedded systems, security is a crucial issue.

10 More particularly, during the development of embedded systems, vulnerabilities may be introduced. Examples include the buffer overflow technique or the integer overflow technique.

 Such vulnerabilities could allow an attacker, in a phase of using the developed embedded system, to recover confidential data, to temporarily or permanently modify the
15 behavior of the embedded system, or even make the embedded system inoperative.

 It is known how to limit the presence of vulnerabilities by the use of a plurality of techniques including unit testing, fuzzer testing, static code analysis, dynamic code analysis or formal proof.

 In addition to the limitation tools, it is also possible to use techniques for detecting
20 the exploitation of vulnerabilities.

 An example of such a technique is the detection of unintended modification of the stack. Stack canary and shadow stacks are two techniques for detecting buffer overflows in the stack.

 However, none of the techniques can detect all vulnerabilities.

25 There is thus a need for a method of controlling an embedded calculator, serving for detecting a larger set of vulnerabilities.

 To this end, the description describes a method for controlling an embedded calculator adapted to control a critical system, the embedded calculator comprising:

- a single kernel suitable for implementing a set of operations,
- 30 - hardware components to which the kernel has access in order to implement the set of operations, and
- a control unit, the control unit being apt to control the access of the kernel to the hardware components, the control unit being isolated from the kernel,

 the control method including a step of control by the control unit of the accesses of
35 the kernel to the hardware components.

According to particular embodiments, the control method has one or a plurality of the following features, taken individually or according to all technically possible combinations:

- the embedded calculator further includes a memory management unit, the control step including the control of the memory management unit.

- the method further includes the steps of:

- analysis of sections of a binary obtained by compilation of the kernel, and

- deduction of access to the hardware components by the kernel during normal operation of the kernel, in order to obtain an access configuration,

the access control step taking into account the access configuration.

- the method includes the generation of a binary, the analysis and deduction steps being implemented using the generated binary.

- during the deduction step, the access configuration corresponds to the minimum number of accesses to the hardware components serving the normal operation of the kernel.

- the control method further includes a step of blocking the modification of the kernel.

- the hardware components include a calculation unit, the method being implemented at the start-up of the calculation unit.

- privilege levels for access to hardware components are defined, the privilege level of the control unit being strictly higher than the privilege level of the kernel.

The description further describes an embedded calculator suitable for controlling a critical system, the embedded calculator comprising:

- a single kernel suitable for implementing a set of operations,

- hardware components to which the kernel has access in order to implement the set of operations, and

- a control unit, the control unit being apt to control the access of the kernel to the hardware components, the control unit being isolated from the kernel.

The description further relates to a vehicle including an embedded calculator such as described hereinabove.

In the present description, the expression "suitable for" means equally well "apt to" or "configured for".

The features and advantages of the invention will appear upon reading the following description, given only as an example, but not limited to, and making reference to the enclosed drawings, wherein:

- Figure 1 is a schematic representation of an example of vehicle including an embedded calculator having a plurality of units, and

- Figure 2 is a schematic representation of the operation of a calculator illustrating the interactions between the units represented in Figure 1.

A vehicle 10 is schematically illustrated in Figure 1.

5 The vehicle 10 is any type of vehicle and could be, more particularly, a land, air or sea vehicle.

The vehicle 10 includes a set 12 of systems among which non-critical systems 14 and critical systems 16.

A system is considered a critical system when a failure of the system jeopardizes the operation of the vehicle 10.

10 For example, the propulsion system or the detection systems are critical systems 16.

The vehicle 10 also comprises an embedded calculator 18 suitable for controlling the critical systems 16.

15 Hereinafter, in order to simplify the reading, the embedded calculator 18 is simply called the calculator 18.

Because of such a control, the calculator 18 is a critical system.

As such, the calculator 18 satisfies a relatively high level of security compared with a standard processor that does not meet security needs.

20 As shown diagrammatically in Figure 1, the calculator 18 includes a single kernel 20, hardware components 22, a memory management unit 24 and a control unit 26.

There is one kernel 20 but said kernel is used to handle a plurality tasks. The above is represented very schematically in Figure 2 by a set of superposed sheets with the reference sign 21.

As such, the kernel 20 is suitable for implementing a set of operations.

25 Thereby, in the following, the tasks or operations that the kernel 20 is apt to implement are associated with the reference sign 21.

The kernel 20 is an element retaining a small number of fundamental functions. All the functions usually provided by monolithic kernels are then provided by elements external to the kernel 20.

30 As a result, it is possible to obtain a small kernel with better security, the security being required for embedded applications.

Typically, a kernel corresponds to less than 50,000 lines of code while a monolithic kernel corresponds to several million lines of code.

35 The hardware components 22 are components to which the kernel 20 has access in order to implement the set of operations 21.

As an illustration only, in the case of Figure 1, it is assumed that the hardware components 22 are a memory 28 and a calculation unit 30.

The functions of the memory 28 and of the calculation unit 30 are herein standard functions for the domain of the embedded system, namely storing data and performing operations 21, respectively.

The kernel 20 can access the memory 28 in three ways:

- read-only, or
- read/write, or
- execute (which uses the presence of a register giving the position of the next instruction to be executed).

The memory management unit 24 is used for controlling memory accesses.

The memory management unit 24 is more often designated by the abbreviation MMU.

The control unit 26 is apt to control the access of the kernel 20 to the hardware components 22.

As will be subsequently described, in the example described the control unit 26 controls the accesses by a control of the memory management unit 24.

The control unit 26 is isolated from the kernel 20 and the tasks 21 the kernel 20 performs.

Isolation means herein that the control unit 26 operates completely independently of the kernel 20.

Furthermore, the control unit 26 has a privilege level of access to the hardware components 22 that is strictly higher than the privilege level of the kernel 20.

Preferentially, the control unit 26 has the highest privilege level.

A good control of the kernel 20 by the control unit 26 results therefrom.

On the other hand, the lower privilege level of the kernel guarantees that the kernel 20 cannot modify the control unit 26.

The control unit 26 corresponds herein to a code of small size, typically less than 1000 lines.

The operation of the calculator 18 is now described with reference to the schematic representation shown in Figure 2 corresponding to the implementation of a method for controlling the calculator 18.

The control method includes a first phase implemented by an analysis unit 32, schematically represented by a square in Figure 2.

It is assumed herein that the kernel 20 is in progress, i.e. that the first phase corresponds to the development/refinement phase of the kernel 20 before the deployment thereof in the field.

When the kernel 20 is compiled (see element C1 in Figure 2), a binary 34 of the
5 program is generated.

Such binary is more often referred to by the corresponding English name "binary". The binary is a file organized into predefined sections, the sections gathering, e.g., instructions or read-only data (the constants).

The binary 34 describes, for the kernel 20, the interactions between the kernel 20
10 and the exterior of the kernel 20.

The analysis unit 32 exploits the binary 34 for analyzing the different sections of the binary 34 in order to generate the access rights associated with the different memory zones according to the data stored therein (constants, executable or the stack)

According to the embodiment described the analysis unit 32 calculates the minimum
15 number of accesses to the different memory zones serving the normal operation of the kernel 20 and the tasks 21 that the kernel carries out.

In this way it is possible in particular to reduce the amount of usable memory to the strict necessary. Such a restriction makes it possible to limit the risks of leakage of confidential data in memory and the associated tests or erasure.

20 Such operation is schematically represented, in Figure 2, by the element 36.

The analysis unit 32 is thereby a tool for generating automatic security policy configuration. As such, the tool is apt to analyze the binary 34 in order to identify the security properties associated with each standard section of the binary 34.

On the basis of such identification, the analysis unit 32 is apt to generate an access
25 configuration 36.

The access configuration 36 can be used in a standard format for making possible, if appropriate, a manual configuration supplement.

More particularly, a manual configuration supplement could consist of defining the security properties associated with the non-standard sections.

30 The access configuration 36 is then converted into a configuration file 40 that can be interpreted by a compiler.

The configuration file 40 gathers the authorized accesses to the memory 28.

The method then comprises a second compilation step (element C2 in Figure 2) corresponding to the compilation of the control unit, the binary-program interface and the
35 access configuration.

The knowledge of the accesses authorized to the hardware components 22 by the control unit 26 enables the latter to control the memory management unit 24.

More precisely, the control unit 26 will require the memory management unit 24 to operate according to rules corresponding to the authorized accesses of the generated configuration.

In order to increase security, the control method further includes a step of blocking the modification of the kernel 20.

The above means that the control unit 26 is used for making the code zones corresponding to the kernel 20 non-modifiable.

A calculator 18 is thereby obtained, for which the control unit 26 is thereby apt to detect any abnormal access request at the memory management unit 24 during the second phase corresponding to the operation of the calculator 18.

As a result, buffer overflow or integer overflow techniques can be detected.

The control method described serves thereby to obtain better security for the calculator 18.

The control has the specificity of being carried out during all the obtaining phases of the calculator 18 and not only during the operational phase.

Indeed, the control is ensured as soon as it is possible to compile the kernel 20.

Therefore, such is already the case during the development phase of the kernel 20.

Furthermore, the same control unit 26 is used during both development and use, which limits the possibility that an attacker exploits a vulnerability of a kernel 20.

The presence of the control unit 26 makes it possible to obtain a calculator 18 compatible with the constraints of the embedded.

Other embodiments of the method can be considered while guaranteeing such good level of security.

More particularly, the method can include a step of creating a log recording all unauthorized accesses.

As a result, the monitoring of the correct operation of the calculator 18 is improved.

It is also indispensable that the control step be implemented at the start-up of the computing unit 30, so as to ensure that the control is always effective.

When the hardware components 22 include components other than the components shown in Figure 1, the access configuration 36 further comprises elements for preventing the components from making modifications or bypassing the controls performed by the memory management unit 24.

CLAIMS

1. A method for controlling an embedded calculator (18) adapted to control a critical system (16), the embedded calculator (18) comprising:

- a single kernel (20) adapted to implement a set of operations (21),
- hardware components (22, 28, 30) to which the kernel (20) has access in order to implement the set of operations (21), and
- a control unit (26), the control unit (26) being adapted to control the access of the kernel (20) to the hardware components (22, 28, 30), the control unit (26) being isolated from the kernel (20),

the control method including a step of control by the control unit (26) of the accesses of the kernel (20) to the hardware components (22, 28, 30).

2. The method for controlling according to claim 1, wherein, the embedded calculator (18) further includes a memory management unit (24), the step of control including the control of the memory management unit (24).

3. The method for controlling according to claim 1 or 2, wherein the method further comprises the steps of:

- analysis of the sections of a binary (34) obtained by compilation of the kernel (20), and
- deduction of accesses to the hardware components (22, 28, 30) by the kernel (20) during normal operation of the kernel (20), in order to obtain an access configuration, the step of control of the accesses taking into account the access configuration.

4. The method for controlling according to claim 3, wherein the method includes the generation of a binary (34), the analysis and deduction steps being implemented using the generated binary.

5. The method for controlling according to claim 3 or 4, wherein, in the step of deduction, the access configuration corresponds to the minimum number of accesses to the hardware components (22, 28, 30) serving the normal operation of the kernel (20).

6. The method for controlling according to any one of claims 1 to 5, wherein the method for controlling further includes a step of blocking the modification of the kernel (20).

7. The method for controlling according to any one of claims 1 to 6, wherein the hardware components (22, 28, 30) include a computing unit (30), the method being implemented at the start of the computing unit (30).

5 **8.** The method for controlling according to any one of claims 1 to 7, wherein privilege levels for accessing the hardware components (22, 28, 30) are defined, the privilege level of the control unit (26) being strictly higher than the privilege level of the kernel (20).

10 **9.** An embedded calculator (18) adapted to control a critical system (16), the embedded calculator (18) comprising:

- a single kernel (20) adapted to implement a set of operations (21),
- hardware components (22, 28, 30) to which the kernel (20) has access in order to implement the set of operations (21), and
- 15 - a control unit (26), the control unit (26) being adapted to control the accesses of the kernel (20) to the hardware components (22, 28, 30), the control unit (26) being isolated from the kernel (20).

10. A vehicle (10) including an embedded calculator (18) according to claim 9.