

user of the detection of an abnormal structure and/or behavior of the active stack; analyzing the operation of the active stack to detect the second party that originated the SCR breach operation; and informing other fellow agents.

[0035] In order to increase the efficiency of the system of the invention, it comprises a plurality of sensors for detecting and eliminating SCR breach operations by a second party within the memory space allocated to a first party, in a multi-tasking system.

[0036] According to one embodiment of the invention a sensor comprises a plurality of probes implanted each within one stack.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0037] In the drawings:

[0038] **FIG. 1A** illustrates the manner of activating SCRs by a conventional multi-tasking operating system;

[0039] **FIG. 1B** schematically illustrates the architecture of the extensible chain, according to the prior art;

[0040] **FIG. 1C** illustrates how a sensor of the invention is activated to protect a plurality of SCR stacks;

[0041] **FIG. 2A** schematically illustrates the main components of a sensor, according to a preferred embodiment of the present invention;

[0042] **FIG. 2B** schematically illustrates the placement of public sensors and private sensors, according to an embodiment of the present invention;

[0043] **FIG. 2C** illustrates the interaction between the monitoring probe and the sensor main unit according to a preferred embodiment of the present invention;

[0044] **FIG. 3** is a flow diagram showing the evaluation of a suspect by its module-name;

[0045] **FIG. 4A** is a flow diagram showing the evaluation on a process level whether an SCR is added to a process;

[0046] **FIG. 4B** is a flow diagram showing the evaluation on the chain level whether an SCR is added to an SCR chain;

[0047] **FIG. 4C** shows handling the occurrence of a replacement of a procedure address in a given process which may indicate a previous unauthorized addition of an SCR to an SCR chain used by the process;

[0048] **FIG. 4D** shows a check of a degradation of execution performance or an indication of an unusual activity in a given process which may indicate a previous unauthorized addition of an SCR to an SCR chain used by the process;

[0049] **FIG. 4E** is a flow diagram showing the evaluation of a change, wherein a new, unexpected thread is being created within some process's context. Where this is not an expected behavior of the process, this may indicate a previous unauthorized addition of an SCR to an SCR chain used by the process; and

[0050] **FIG. 5** is a flow diagram showing the operation of the sensor's decision unit.

#### DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0051] Throughout this specification, the following definitions are employed:

[0052] **Handler:** in the context of this application, the term handler is used herein in a free manner to denote (I) a software routine that performs a particular task on the fly or (II) a package of some routines with a common destination. Each layer comprising of at least one handler.

[0053] **SCR:** Shared Code Resource—in the context of this application, an SCR is an executable program module that perform some particular functions on behalf of other SCRs, independent executables, or the OS itself.

[0054] **Computerized system:** in the context of this application, refers to one or more machines that operate by an OS.

[0055] **Task:** in the context of this application, a task is the running session of a program, an application, or some other piece of code on a computerized system.

[0056] **Process:** action operating in a multi-tasking system which uses part of the computerized system resources. Under Windows, each process has at least one thread of execution (see below).

[0057] **User-mode process:** Processes that run in the so-called 'user-mode' are assigned a virtual private address space, and the OS maps between physical memory addresses and each process address space. The 'memory-space' of each user-mode process should be isolated. This requirement arises from stability considerations. For example, a faulty operation of one user process should not crash the operation of another process or the whole system. Another reason for this isolation is the need for security, as discussed above. However, as will be shown hereinafter, this isolation can be broken, resulting in a breach of security.

[0058] **Thread:** a thread is a mechanism that enables concurrent flow of execution within a given process. It can utilize multiprocessor machines when available, or merely harness CPU cycles. Threads are useful for tasks that require concurrent processing, for tasks that need user interaction while doing CPU-intensive activity, and for 'server' programs where new threads are launched for each incoming request, to smoothen and isolate concurrent requests from multiple client applications and, possibly, multiple users.

[0059] **Name:** Throughout this document, when the term "SCR name" or "module name" are used, it is referred to the name of the disk-file that holds the image of an SCR or a so-called module, including the whole sequence of global path, local name, extension, etc. This is important because a well-known technique for diverting the expected functionality of a given SCR, at least under Windows™, is to plant an SCR with the same local name of the original SCR, but at the path where the OS is likely to search first. Typically the new SCR will be placed at the same folder of the target program (which expects to use the original SCR) executable file, while the original SCR rests at a common folder where the OS keeps such SCRs.

[0060] **FIG. 1A** illustrates the manner of activating SCRs by a conventional multi-tasking operating system. The operating system (not indicated in this figure) contains a library