

tools and intrusion-detection systems, and makes the search for accountability more difficult.

[0015] In general, memory-space breaching seems as an appropriate technique for an invader, whose goal is not mere vandalizing of an invaded site. Sophisticated contamination of the victim's valued information resources would be a possible goal. Altering the behavior of unaware information-security systems through their user-mode components is another goal, so is eavesdropping or stealing information, to mention just a few.

[0016] For example, a ubiquitous OS like MS Windows™ (Microsoft Corporation, USA) gives a program the ability to order the OS to extend, on the fly, SCR stacks, wherein each SCR provides a particular level of functionality. In many cases this goes on without alerting the user of this OS. The extension is done by adding at least one SCR to the chain, wherein this SCR may serve more than one application or process concurrently.

[0017] There are several mechanisms in the OS that might be extended by additional SCRs. The following are examples for some well known in the art of such mechanisms:

[0018] (i) windows-messages that may be hooked;

[0019] (ii) video and audio Compressor/De-Compressor (codecs);

[0020] (iii) Windows Open Services Architecture (WOSA) stacks, which is a collective term for a variety of programming interfaces from Microsoft designed to provide application interoperability across the Windows environment. An example for WOSA is the Windows Socket (Winsock), which is a Windows interface to a communications protocol over the Internet; and

[0021] (iv) There are more WOSA mechanisms like ODBC and MAPI. Furthermore, there is the infrastructure for the 'Component Services' of Windows. This list is by no means a complete list of all the vulnerable service chains in a modern OS, but only a list of examples.

[0022] The art has not yet provided satisfactory protection means for detecting and/or preventing such inter-process memory breaches in multitasking OS.

[0023] It is an object of the present invention to improve the security in multi-users and multitasking systems.

[0024] It is another object of the present invention to provide a method and system for detecting an illegal action of penetrating a memory space of one process by another process.

[0025] It is further object of the present invention to provide a method and system for detecting a process that initiates such penetrating action.

[0026] It is a still further object of the present invention to provide a method and system for freezing the action of the invader and/or the invaded processes, and alerting on such illegal action.

[0027] Other objects and advantages of the invention will become apparent as the description proceeds.

SUMMARY OF THE INVENTION

[0028] The invention relates to a method 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, which comprises: (a) pre-recording by the first party within a knowledge base the structure and/or behavior of an SCR stack; (b) implanting within the SCR stack a dedicated SCR for reporting on the structure and/or behavior of said SCR stack when the SCR stack is activated; (c) when the SCR stack is activated, comparing the data reported by the dedicated SCR with the pre-recorded stack structure and/or behavior; (d) whenever non-matching in the structure and/or behavior is found, ceasing the activity of the activated stack, and alerting.

[0029] Preferably the comparison of structure comprises verification of one or more of the following: the number of SCRs within the stack; the chain order of the SCRs within the stack; the time-stamps of the SCRs within the stack; the names of the SCRs within the stack; a signature of each SCR within the stack; the number of bits of each SCR within the stack; a checksum of each SCR within the stack; the physical path and name of each SCR within the stack.

[0030] Preferably the comparison of behavior comprises verification of one or more of the following: duration of performance of the stack, and/or each SCR within the stack; the I/O devices and/or addresses to which a communication is made when the stack is activated by a specific process.

[0031] According to one embodiment of the invention the SCR breach operation is carried out by means of implanting SCRs within a shared stack by the second party. According to another embodiment of the invention the SCR breach operation is carried out by means of implanting or manipulating by the second party an SCR within a shared stack supposed to be activated by the first party, and wherein the SCR implanted or manipulated by the second party is designed to perform operations within the memory space exclusively allocated to the first party.

[0032] Detecting and eliminating SCR breach operations by a second party within the memory space allocated to a first party, in a multi-tasking system is performed, preferably, with respect to each stack supposed to be activated by the first party.

[0033] Preferably, the stack behavior is checked independent of the process that activating it and/or the stack behavior is checked specifically with respect to the process that activating it.

[0034] The invention further relates to a sensor 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, which comprises: (a) at least one probe implanted within a stack by the first party, for reporting on the structure and/or behavior of the SCR stack, when the SCR stack is activated; (b) a knowledge base for containing information relating to the structure and/or behavior of the stack, when activated; (c) a comparing unit for comparing information relating to the stack structure and/or behavior as reported by the probe, with information recorded in the database; and (d) a decision unit capable of initiating one or more of the following operations, if abnormal structure and/or behavior of the active stack is detected in step c: ceasing operation of the active stack; alerting the