CLAIMS

1. A method implemented in a computer comprising a processor (302), memory (308) accessible by the processor, and computer program instructions stored in the memory and executable by the processor, the method characterised as comprising:

creating (202) a template system call page table (126), the template system call page table comprising mappings (128) of kernel code and data required to perform an entry to a kernel mode upon execution of a system call, a table with pointers to a set of system call handlers, and entry code of each system call handler in an entry page table (122) of an operational system call page table (104), a populated kernel entry page table (120) and an empty user-space page table (118);

in response to a user-space process being created (204), copying the template system call page table to create the operational system call page table for the user-space process in a process state representation and populating the empty user-space table of the operational system call page table with user-space mappings in the same way as a normal process page table, wherein the operational system call page table comprises a plurality of kernel-space address ranges allowing execution of kernel code and access to kernel data;

executing (206) a system call, by the user-space process;

when the user-space process makes the system call, switching to kernel mode and invoking a generic system call entry routine, the invoking switching a page table pointer register to use the operational system call page table and to execute a system call handler of the operational system call page table;

in response to the system call handler accessing memory comprising kernel-space kernel code and/or data, generating a page fault;

verifying, by a security block (124), whether the requested memory access is allowed;

in response to the access being determined to be allowed, adding a mapping of the requested memory to the operational system call page table;

in response to execution of the system call handler ending (210), transferring control to the generic system call entry routine; cleaning up the operational system call page table (104); and removing mappings added during the execution of the system call handler.

2. The method of claim 1, wherein the computer system operating system is the Linux operating system.

3. The method of claim 1, further comprising:

when a page fault is determined not to be allowed, terminating execution of the user-space process that made the system call that attempted to access kernel-space memory.

4. A system comprising a processor (302), memory (308) accessible by the processor, and computer program instructions stored in the memory and executable by the processor to perform:

creating (202) a template system call page table (126), the template system call page table comprising mappings (128) of kernel code and data required to perform an entry to a kernel mode upon execution of a system call, a table with pointers to a set of system call handlers, and entry code of each system call handler in an entry page table (122) of an operational system call page table (104), a populated kernel entry page table 120 and an empty user-space page table (118);

in response to a user-space process being created (204), copying the template system call page table to create the operational system call page table for the user-space process in a process state representation and populating the empty user-space table of the operational system call page table with user-space mappings in the same way as a normal process page table, wherein the operational system call page table comprises a plurality of kernel-space address ranges allowing execution of kernel code and access to kernel data;

executing (206) a system call, by the user-space process;

when the user-space process makes the system call, switching to kernel mode and invoking a generic system call entry routine, the invoking switching a page table pointer register to use the operational system call page table and to execute a system call handler of the operational system call page table;

in response to the system call handler accessing memory comprising kernel-space kernel code and/or data, generating a page fault;

verifying, by a security block (124), whether the requested memory access is allowed;

in response to the access being determined to be allowed, adding a mapping of the requested memory to the operational system call page table;

in response to execution of the system call handler ending, transferring control to the generic system call entry routine; cleaning up the operational system call page table (104); and removing mappings added during the execution of the system call handler.

5. The system of claim 4, wherein the computer system operating system is the Linux operating system.

6. The system of claim 4, further comprising:

when a page fault is determined not to be allowed, terminating execution of the user-space process that made the system call that attempted to access kernel-space memory.

7. A computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions executable by a computer, to cause the computer to perform a method comprising any of the steps of claims 1 to 3.