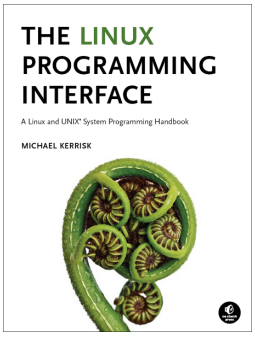
man7.org
Training and Consulting

Linux Security and Isolation APIs

Course code: M7D-SECISOL02

This course provides a deep understanding of the low-level Linux features (set-UID/set-GID programs, capabilities, namespaces, cgroups, and seccomp) used to implement privileged applications and build container, virtualization, and sandboxing technologies. Detailed presentations coupled with carefully designed practical exercises provide participants with the knowledge needed to understand, design, develop, and administer such applications. (The course does *not* cover administering container systems such as Docker and LXC, but by completion of the course you will have a good understanding of various aspects of the underlying implementation and operation of such systems.)



Audience and prerequisites

The primary audience comprises designers and programmers building privileged applications, container applications, and sandboxing applications. Systems administrators who are managing such applications are also likely to find the course of benefit.

Participants should have working knowledge of the fundamental system programming topics covered in the *Linux System Programming Fundamentals* course (M7D-SPINTRO01). This includes file descriptors, file I/O using system calls, signals, and the system calls that define the lifecycle of a process (*fork()*, *exec()*, *wait()*, *exit()*).

Participants should have a good reading knowledge of the C programming language and some programming experience in a language suitable for completing the course exercises (e.g., C, C++, Go, Rust).

Course materials

- A course book (written by the trainer) that includes all course slides and exercises
- A source code tarball containing all of the (many) example programs written by the trainer to accompany the presentation

Course duration and format

Four days, with around 40% of the course time devoted to practical sessions.

Course inquiries and bookings

For inquiries about courses and consulting, you can contact us in the following ways:

- Email: training@man7.org
- Phone: +49 (89) 2155 2990 (German landline)

Prices, dates, and further details

For course prices, upcoming course dates, and further information about the course, please visit the course web page, *http://man7.org/training/sec_isol_apis/*.

About the trainer

Michael Kerrisk has a unique set of qualifications and experience that ensure that course participants receive training of a very high standard:

- He has been programming on UNIX systems since 1987 and began teaching UNIX system programming courses in 1989.
- He is the author of *The Linux Programming Interface*, a 1550-page book widely acclaimed as the definitive work on Linux system programming.
- He is actively involved in Linux development, working with kernel developers on testing, review, and design of new Linux kernel–user-space APIs.
- Since 2004, he has been the maintainer of the Linux *man-pages* project, which provides the manual pages documenting the Linux kernel–user-space and GNU C library APIs.

Linux Security and Isolation APIs: course contents in detail

Topics marked with an asterisk (*) may be covered, if time permits.

1. **Background (covered as needed) (*)**
   - Process credentials
   - Signals and signal handlers
   - Process lifecycle (*fork()*, *execve()*, *exit()*, *waitpid()*)
   - The `/proc` filesystem
2. **Cgroups**
   - Overview/purpose of cgroups
   - Hierarchies and controllers
   - Populating a cgroup
   - Release notification
   - A survey of the cgroups v1 controllers
   - `/proc` files
3. **Cgroups v2**
   - Problems with cgroups v1; rationale for v2
   - Cgroups v2 controllers
   - Enabling and disabling controllers
   - Organizing cgroups and processes
   - Release notification
   - Delegation
   - Thread mode (*)
4. **Seccomp**
   - Introduction and history
   - Seccomp filtering and BPF
   - The BPF virtual machine and BPF instructions
   - Checking the architecture
   - Filter return values
   - BPF programs
   - Discovering the system calls made by a program
   - *libseccomp*
   - Caveats regarding the use of seccomp
   - Applications, tools, and further information
5. **Privileged programs**
   - Set-UID and set-GID programs
   - Changing process credentials
   - Guidelines for writing privileged programs
6. **Capabilities**
   - Process and file capabilities (permitted, effective, and inheritable)
   - Viewing and setting file capabilities from the shell
   - Text-form capabilities
   - Transformation of capabilities by *execve()*
   - Capability bounding set
   - Root, UID transitions, and securebits (*)
   - Capabilities APIs (*)
   - Ambient capabilities
   - Problems with capabilities
7. **Namespaces**
   - Overview
   - Namespace types
   - UTS namespaces
   - Mount namespaces; shared subtrees
   - IPC namespaces
   - Cgroup namespaces
   - Network namespaces (overview)
   - PID namespaces
   - User namespaces (overview)
   - Namespaces APIs: *clone()*, *setns()*, *unshare()*, and *ioctl()*, `/proc/PID/ns/*`
8. **User namespaces in depth**
   - UID and GID mappings
   - *execve()* and UID 0 semantics
   - User namespaces and capabilities
   - Combining user namespaces with other namespace types
   - User namespaces and capabilities revisited
9. **Mount namespaces (*)**
   - Introduction to mount namespaces
   - Shared subtrees
   - Bind mounts
   - Peer groups and mount propagation
   - Shared and private mounts
   - Slave mounts
   - Unbindable mounts
10. **Network namespaces (*)**
    - Overview
    - Resources isolated by network namespaces
    - Virtual networking devices
    - The *ip netns* command
    - Configuring network namespaces
    - Using physical network devices with network namespaces
    - Use cases for network namespaces