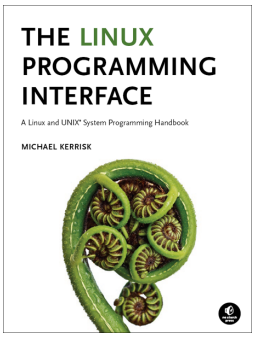
Linux Security and Isolation APIs

Course code: M7D-SECISOL01

This course provides a deep understanding of the low-level Linux features (set-UID/set-GID programs, capabilities, namespaces, 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 a good reading knowledge of the C programming language and solid programming experience in a language suitable for completing the course exercises (e.g., C, C++, Go).

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

Two or three days (depending on course content), 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 1500-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

Because of time constraints, topics marked with an asterisk (*) are normally covered only in the three-day version of this course. Topics marked with two asterisks (**) may be covered in the three-day version of the course, if time permits.

1. **Background (covered as needed)**
   - Process credentials
   - Process lifecycle (*fork()*, *execve()*, *exit()*, *waitpid()*)
   - The `/proc` filesystem
2. **Privileged programs**
   - Set-UID and set-GID programs
   - Changing process credentials
   - Guidelines for writing privileged programs
3. **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
4. **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()*
5. **Mount namespaces (**)**
   - Introduction to mount namespaces
   - Shared subtrees
   - Bind mounts
   - Peer groups
   - Shared and private mounts
   - Slave mounts
   - Unbindable mounts
6. **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
7. **Seccomp**
   - Introduction and history
   - BPF (Berkeley Packet Filter)
   - Constructing seccomp filters
   - BPF programs
   - *libseccomp* (*)
   - Applications, tools, and further information
8. **Cgroups (*)**
   - Overview/purpose of cgroups
   - Cgroup filesystem
   - Hierarchies and controllers
   - Populating a cgroup
   - Resource controllers (PID, CPU, memory, etc.)
   - Cgroups v2 (rationale, design changes, single unified hierarchy, delegation)