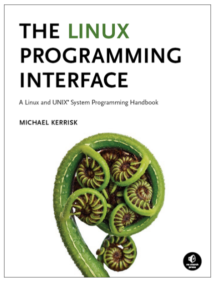
man7.org
Training and Consulting

Linux Security and Isolation APIs Fundamentals

Course code: M7D-SISINTRO01

This course provides an introduction to the low-level Linux features–set-UID programs, capabilities, namespaces, control groups (v2), and seccomp–used to build containers and sandboxing systems.


THE LINUX PROGRAMMING INTERFACE
A Linux and UNIX System Programming Handbook
MICHAEL KERRISK

Audience and prerequisites

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

Participants should have working knowledge of the fundamental system programming topics covered in the *Linux System Programming Essentials* (M7D-SPESS01) course. This includes file descriptors and file I/O, signals, and the process lifecycle (*fork()*, *exec()*, *wait()*, *exit()*). In addition, participants should have a reading knowledge of the C programming language. (Note, however, that the course exercises do not require writing any programs.)

Related courses

The *Linux Security and Isolation APIs* (M7D-SECISOL02) course covers the same topics as this course, but in greater depth.

Course materials

- Course books (written by the trainer) that include all slides and exercises presented in the course
- An electronic copy of the trainer's book, *The Linux Programming Interface*
- Numerous example programs written by the course trainer

Course duration and format

Two 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/secisolintro/`.

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 acclaimed as the definitive work on Linux system programming.
- He has been actively involved in Linux development, working with kernel developers on testing, review, and design of new Linux kernel–user-space APIs.
- Since 2000, he has been the involved in the Linux *man-pages* project, which provides the manual pages documenting Linux system calls and C library APIs, and was the project maintainer from 2004 to 2021.

Linux Security and Isolation APIs Fundamentals: course contents in detail

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

1. **Course Introduction**
2. **Classical Privileged Programs**
   - A simple set-user-ID program
   - Saved set-user-ID and and saved set-group-ID
   - Changing process credentials
   - A few guidelines for writing privileged programs
3. **Capabilities**
   - Process and file capabilities
   - Permitted & effective capabilities
   - Setting & viewing file capabilities
   - Capabilities-dumb and capabilities-aware applications
   - Text form capabilities
   - Capabilities and *execve()*
   - The capability bounding set
   - Capabilities and UID transitions
   - Summary remarks
4. **Capabilities: Further Topics**
   - Capabilities, UID 0, and *execve()*
   - Programming with capabilities (*)
5. **Namespaces**
   - An example: UTS namespaces
   - Namespaces commands
   - Namespaces demonstration (UTS namespaces)
   - Namespace types and APIs
6. **Mount Namespaces and Shared Subtrees**
   - Mount namespaces
   - Shared subtrees
7. **PID Namespaces**
8. **Namespaces APIs**
   - API Overview
   - Creating a child process in new namespaces: *clone()*
   - `/proc/PID/ns`
   - Entering a namespace: *setns()*
   - Creating a namespace: *unshare()*
   - PID namespaces idiosyncrasies (*)
9. **User Namespaces**
   - Overview of user namespaces
   - Creating and joining a user namespace
   - User namespaces: UID and GID mappings
   - User namespaces, *execve()*, and user ID 0
   - Use cases
   - Combining user namespaces with other namespaces
10. **User Namespaces and Capabilities**
    - User namespaces and capabilities
    - What does it mean to be superuser in a namespace?
11. **Cgroups: Introduction**
    - Preamble
    - What are control groups?
    - An example: the `pids` controller
    - Creating and destroying cgroups
    - Populating a cgroup
    - Enabling and disabling controllers
12. **Cgroups: A Survey of the Controllers**
    - The `cpu`, `memory`, `freezer`, and `pids` controllers
    - Other controllers (*)
13. **Seccomp (*)**
    - Seccomp filtering and BPF
    - The BPF virtual machine and BPF instructions
    - BPF filter return values
    - BPF programs
    - Checking the architecture
    - Productivity aids (*libseccomp* and other tools)
    - Applications and further information
