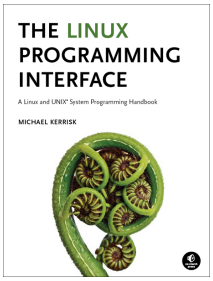
man7.org
Training and Consulting

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.



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

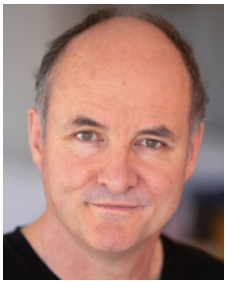
For inquiries about courses and consulting, you can contact us in the following ways:

- Email: training@man7.org
- Phone: +49 (89) 2488 6180 (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 and effective capabilities
   - Setting and viewing file capabilities
   - Capabilities-dumb and capabilities-aware applications
   - Text-form capabilities
4. **Capabilities and *execve()***
   - Capabilities and *execve()*
   - The capability bounding set
   - Summary remarks
5. **Capabilities and UID 0**
   - Capabilities and UID transitions
   - Capabilities, UID 0, and *execve()*
6. **Programming with capabilities (*)**
   - Programming with capabilities
7. **Namespaces**
   - An example: UTS namespaces
   - Namespaces commands
   - Namespaces demonstration (UTS namespaces)
   - Namespace types and APIs
8. **Mount Namespaces and Shared Subtrees**
   - Mount namespaces
   - Shared subtrees
9. **PID Namespaces**
10. **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 (*)
11. **User Namespaces**
    - Overview of user namespaces
    - Creating and joining a user namespace
    - User namespaces: UID and GID mappings
    - Accessing files (and other objects with UIDs/GIDs)
    - Combining user namespaces with other namespaces
    - Use cases
12. **User namespaces, *execve()*, and user ID 0**
    - User namespaces, *execve()*, and user ID 0
13. **User Namespaces and Capabilities**
    - User namespaces and capabilities
    - What does it mean to be superuser in a namespace?
14. **Cgroups: Introduction**
    - Preamble
    - What are control groups?
    - An example: the `pids` controller
    - Creating and destroying cgroups
    - Populating a cgroup
    - Enabling and disabling controllers
15. **Cgroups: A Survey of the Controllers**
    - The `cpu`, `memory`, `freezer`, and `pids` controllers
    - Other controllers (*)
16. **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

The following are some of the **other courses taught by Michael Kerrisk**. Custom courses are also available upon request. Further details on these and other courses can be found at `http://man7.org/training/`. For course inquiries please email training@man7.org or phone +49 (89) 2488 6180 (German landline).

Linux Security and Isolation APIs

Course code: M7D-SECISOL02 (4 days)

Covering topics including control cgroups (cgroups), namespaces (with a deep dive into user namespaces), capabilities, and seccomp (secure computing), this course provides a deep understanding of the low-level Linux features used to design, build, and troubleshoot container, virtualization, and sandboxing frameworks. [This course is an expanded version of the course described above.]

Linux/UNIX Network Programming

Course code: M7D-NWP03 (3 days)

This course covers sockets programming (both UNIX and Internet domain sockets), and the use of relevant I/O techniques for working with sockets (*poll()*, *epoll*, nonblocking I/O). In addition, we look at the TCP/IP protocol stack (including details of TCP such as the 3-way handshake and the TCP state machine), the use of monitoring and tracing tools (*ss*, *netstat*, and *tcpdump*/*wireshark*), and raw sockets.

Linux/UNIX System Programming

Course code: M7D-LUSP01 (5 days)

This course covers the APIs used to build system-level applications on Linux and UNIX systems ranging from embedded processors to enterprise servers. The presentations and practical exercises provide participants with the knowledge needed to write complex system, network, and multithreaded applications. Topics include: file I/O; signals; process creation and termination; program execution; POSIX threads; interprocess communication, and I/O multiplexing (*poll()*, *epoll*).

Building and Using Shared Libraries on Linux

Course code: M7D-SHLIB04 (2.5 days)

This course describes how to design, build, and use shared libraries on Linux. Topics include: fundamentals of library creation and use; shared library versioning; symbol resolution; library search order; executable and linking format (ELF); dynamically loaded libraries; controlling symbol visibility; and symbol versioning.