

Szkolenie: The Linux Foundation
LFD440 Linux Kernel Debugging and Security

FORMA SZKOLENIA	MATERIAŁY SZKOLENIOWE	CENA	CZAS TRWANIA
Stacjonarne	Tradycyjne	5800 PLN NETTO*	4 dni
Stacjonarne	Tablet CTAB	6200 PLN NETTO*	4 dni
Metoda dlearning	Tradycyjne	5800 PLN NETTO*	4 dni
Metoda dlearning	Tablet CTAB	5800 PLN NETTO*	4 dni

* (+VAT zgodnie z obowiązującą stawką w dniu wystawienia faktury)

LOKALIZACJE

Kraków - ul. Tatarska 5, II piętro, godz. 9:00 - 16:00

Warszawa - ul. Bielska 17, godz. 9:00 - 16:00

DOSTĘPNE TERMINY

2019-09-24 | 4 dni | Warszawa

2019-09-24 | 4 dni | Warszawa (Promocja)

2019-11-26 | 4 dni | Kraków

2019-11-26 | 4 dni | Kraków (Promocja)

Cel szkolenia:

Ten 4-dniowy kurs zapoznaje doświadczonych programistów z metodami i wewnętrzną infrastrukturą jądra Linuxa. Kurs skupia się na ważnych narzędziach używanych do debugowania i monitorowania jądra oraz na tym, jak są wdrażane i kontrolowane funkcje bezpieczeństwa. Zawiera obszernie praktyczne ćwiczenia i demonstracje, których celem jest dostarczenie niezbędnych narzędzi do opracowania i debugowania kodu jądra Linuxa.

Plan szkolenia:

- Wprowadzenie
 - Cele
 - Przedstawienie uczestników
 - The Linux Foundation
 - Linux Foundation Training
 - Dystrybucje Linuxa
 - Platformy

- Przygotowanie systemu
- Używanie i pobieranie maszyny wirtualnej
- Rzeczy zmieniające się w Linuksie
- Dokumentacja i linki
- Rejestracja kursu
- Czynności wstępne
 - Procedury
 - Wersje jądra
 - Źródła jądra i zasotosowanie git
- Jak pracować w projekcie Open Source **
 - Omówienie jak prawidłowo współpracować w takim projekcie
 - Skup się na bezpieczeństwie i jakości
 - Badanie i zrozumienie DNA projektu
 - Dowiedz się nad czym chcesz pracować
 - Identyfikacja opiekunów i ich przepływy pracy i metody
 - Uzyskanie wczesnego wejścia i pracy w otwartym środowisku
 - Przekazuj przyrostowe bity, a nie duże zrzuty kodu
 - Zostaw swoje ego za drzwiami
 - Bądź cierpliwy, rozwijaj relacje długoterminowe, bądź pomocny
- Funkcje jądra
 - Składniki jądra
 - Przestrzeń użytkownika a przestrzeń jądra
 - Co to są wywołania systemowe?
 - Dostępne połączenia systemowe
 - Algorytmy planowania i struktury zadań
 - Kontekst procesu
 - Laboratoria
- Monitorowanie i debugowanie
 - Pakiety Debuginfo
 - Śledzenie i profilowanie
 - sysctl
 - Klucz SysRq
 - Wiadomości oops
 - Debugery jądra
 - debugfs
 - Laboratoria

- System plików proc **
 - Co to jest system plików proc?
 - Tworzenie i usuwanie wpisów
 - Czytanie i pisanie wpisów
 - Interfejs seq_file **
 - Laboratoria
- kprobes
 - kprobes
 - kretprobes
 - SystemTap **
 - Laboratoria
- Ftrace
 - Co to jest ftrace?
 - ftrace, trace-cmd i kernelshark
 - Dostępne narzędzia śledzące
 - Używanie ftrace
 - Pliki w katalogu śledzenia
 - Opcje śledzenia
 - Wyświetlanie za pomocą trace_printk()
 - Trace Markers
 - Czyszczenie bufora
 - trace-cmd
 - Laboratoria
- Perf
 - Czym jest perf?
 - perf stat
 - perf list
 - perf record
 - perf report
 - perf annotate
 - perf top
 - Laboratoria
- Crash
 - Crash
 - Główne komendy
 - Laboratoria

- Główne zrzuty jądra
 - Generowanie zrzutów jądra
 - kexec
 - Konfigurowanie zrzutów jądra
 - Laboratoria
- Wirtualizacja**
 - Co to jest wirtualizacja?
 - Pierścienie wirtualizacji
 - Hypervisors
- QEMU
 - Czym jest QEMU?
 - Emulowane architektury
 - Formaty obrazu
 - Integracja z innymi hypervisor'ami
- Narzędzia do debugowania jądra systemu Linux
 - Wbudowane narzędzia i pomocniki do jądra Linuxa
 - kdb
 - qemu+gdb
 - kgdb: hardware+serial+gdb
 - Laboratoria
- Wbudowany Linux**
 - Systemy wbudowane i czasu rzeczywistego
 - Dlaczego warto korzystać z Linuksa?
 - Tworzenie małego środowiska Linux
 - Linuksy w czasie rzeczywistym
- Powiadamiacze**
 - Czym są powiadamiacze?
 - Struktury danych
 - Wywołania zwrotne i powiadomienia
 - Tworzenie łańcuchów powiadomień
 - Laboratoria
- Skalowanie częstotliwości procesora **
 - Co to jest skalowanie częstotliwości i napięcia?
 - Powiadamiacz
 - Sterowniki
 - Laboratoria

- Netlink Socket **
 - Czym są netlink Sockets?
 - Otwieranie netlink Socket
 - Wiadomości netlink
 - Laboratoria
- Wprowadzenie do bezpieczeństwa jądra systemu Linux
 - Podstawy zabezpieczeń jądra systemu Linux
 - Discretionary Access Control (DAC)
 - POSIX ACLs
 - Możliwości POSIX
 - Przestrzenie nazw
 - Linux Security Modules (LSM)
 - Netfilter
 - Metody kryptograficzne
 - The Kernel Self Protection Project
- Linux Security Modules (LSM)
 - Czym są moduły bezpieczeństwa systemu Linux?
 - Podstawy LSM
 - Wybór LSM
 - Jak LSM działa
 - Przykład LSM: Tomoyo
- SELinux
 - SELinux
 - Ogólny zarys SELinux
 - Tryby SELinux
 - Polityki SELinux
 - Narzędzia kontekstowe
 - SELinux i standardowe narzędzia wiersza poleceń
 - Dziedziczenie i zachowanie kontekstu SELinux **
 - restorecon**
 - semanage fcontext**
 - Korzystanie z SELinux Booleans **
 - getsebool i setsebool**
 - Narzędzia do rozwiązywania problemów
 - Laboratoria
- AppArmor

- Czym jest AppArmor?
- Sprawdzanie statusu
- Tryby i profile
- Narzędzia
- Netfilter
 - Czym jest netfilter?
 - Połączenia w Netfilter
 - Implementacja Netfilter
 - Podłączenie do Netfilter
 - Iptables
 - Laboratoria
- Wirtualny system plików
 - Czym jest wirtualny system plików
 - Dostępne systemy plików
 - Specjalne systemy plików
 - tmpfs
 - ext2/ext3
 - ext4
 - btrfs
 - Wspólny model plików
 - Połączenia systemowe VFS
 - Pliki i procesy
 - Montowanie systemów plików
- Filesystems in User-Space (FUSE)**
 - Czym jest FUSE?
 - Pisanie systemu plików
 - Laboratoria
- Kronikowanie systemu plików**
 - Czym jest kronikowanie systemu plików?
 - Dostępne kronikowane systemy plików
 - Kontrastowe funkcje
 - Laboratoria
- Zakończenie i ankieta oceniająca

** Te sekcje mogą być uznane za częściowo lub w całości jako opcjonalne. Zawierają materiały źródłowe, tematy specjalistyczne lub przedmioty zaawansowane. Instruktor może zdecydować się na ich realizację lub nie, w zależności od doświadczenia w grupie i ograniczeń czasowych.

Wymagania:

Przed rozpoczęciem tego kursu powinieneś:

- Być biegły w języku programowania C.
- Znać podstawowe narzędzia systemu (UNIX), takie jak ls, grep i tar.
- Swobodnie używać edytorów tekstów (np. Emacs, vi itp.).
- Doświadczenie z jakąkolwiek dużą dystrybucją Linuxa jest pomocne, ale nie jest wymagane.
- Ukończyć kurs **LFD420: Linux Kernel Internals and Development** lub mieć równoważne doświadczenie/ wiedzę.

Poziom trudności



Certyfikaty:

Uczestnicy otrzymają **certyfikaty** podpisane przez **The Linux Foundation**.

Prowadzący:

Certyfikowany trener The Linux Foundation.