L-5K Linux Debugging Techniques *: Kernel-mode*

Duration: 3 days	
Pre-requisites	
Mandatory	***Preferable***
Course L-0 “Linux CLI and Shell Scripting” *plus*	Experience developing Linux kernel code with 'C', Course L-3 “Linux Device Drivers” strongly recommended.
Course C-1 “Programming in C” *plus*	
Course level L-1 “Linux Systems Programming”	
Course L-2 “Linux Kernel Internals”	
Below is the Outline TOC (Table Of Contents) document: it presents the (approximate) Day-wise Coverage.	

Day 1

Module 1 : Overview - Using QEMU

What is QEMU?
Usage with a root filesystem and kernel image
- Preparing the Kernel
- Preparing the Root Filesystem (RFS)
- Preparing an Initrd or SDcard image

Lab Assignment: Build a functioning system (kernel + initrd-based root filesystem) using QEMU.

Kernel-Space Code-Based Debugging Techniques

Module 2 : Code-based Techniques for Debugging

Debugging by Printing with the printk
- Loglevels
- How Messages Get Logged
- Turning Messages On and Off
- Rate Limiting
- Where you can and cannot use printk

Ftrace functionality
- Basics
- Tracing using ftrace
- Code-Based tracing

A Header of Convenience

Using **debugfs**
Debugging with ioctl

Day 2

Module 3 : Debugging System Faults

Oops Messages

- Generating a (trivial) Oops
- Analyzing an Oops dump
  - Article: “The foggy crystal ball: Understanding Oopses”
- Generating a mixed source-assembly-machine-language dump
- Kernel Mailing List Oops
- Lab Assignment

System Hangs

- Using the Magic SysRq Facility
- Kernel Panic
  - Setting up your own panic handler using the kernel panic chain notifier mechanism

Module 4 : Kernel-Level Debuggers

Using **gdb** for kernel-space

- gdb and loadable kernel modules

KGDB

- Introduction
- kgdb setup
- Initiating a debugging session on the target system
- Using the kernel debugger kgdb
  - Useful gdb Macros
  - Debugging kernel modules

- Using **QEMU and [k]gdb** for source-level kernel debugging
  - Stand-alone kernel debug
  - Building a root filesystem for Qemu VM
  - Sample Debug Session

KDB

- kdb Kernel Patching, build and installation
- kdb Activation
- kdb Commands
- kdb Debug Session

KDB / KGDB live session demo on the Raspberry Pi hardware board.

objdump

gcc : setting debug flags

- For debug symbolic information
- For mixed source-assembly-machine language listing

Day 3

Module 5 : Kernel-Level Tools

Dynamic Probes – Kprobes & Jprobes

Introduction
Kernel build and setup
Kprobe Interfaces

Probe example with sys_open
Probing at an offset within a function

Kprobes and Loadable Kernel Modules
A basic framework
Usage

Using Jumper Probes (jprobes)

Lab Assigments

Kdump, kexec and crash

The kexec with kdump feature
Tools and kernel installation
Usage Procedure
crash utility
Case Study whitepaper

Kernel Hacking Config Options

Wrap Up