Reverse Engineering Training
Reverse Engineering Training - at a glance

Scope

- 6 days of training - study sessions and hands-on training
- Training focused on basic reverse engineering and basic Assembly
- Focus on the ARM architecture
- Focus on hands-on - guided training and practice

Course Targets

- Basic knowledge on Assembly Programming including writing assembly code in an ARM-based Linux environment
- Fundamentals of the ARM architectures family, focused on ARMv8
- Including aarch32: thumb, thumb-2, arm and aarch64
- Familiarity with ARM instruction set and features
- Gain practical static and dynamic analysis skills for reverse engineering
- An understanding of disassembled code and disassemblers
- Decompilers and their advantages and drawbacks
- Debugging binaries & debuggers work under the hood Dealing with behavior modification
- Working with ghidra/ida according to existing license (no license needed for ghidra)
Course outline

- **Basic Assembly Programming – 2 days**
  - Fundamentals of The ARM Architecture
  - Writing Assembly code in an ARM-based Linux environment
  - ARMv8's features
  - ARM instruction set including different modes
  - Writing shellcode

- **Introduction to Reverse Engineering – 3 days**
  - Reading Disassembly
  - Navigating and orientation in binary files
  - Understanding ABI and common code patterns
  - Using disassembly tools
  - Techniques and best behaviors
  - Understanding binary layout
  - De-compilation & tools
  - Debugging & use in disassembly
  Optional: scripting in IDA/ghidra for reversing

- **Advanced topic in Reverse Engineering – 1 day**
  → some of the topics below according to client selection

  → select up to 2 topics

  - binary patching
  - Combining static and dynamic analysis in a productive manner
  - Anti-reversing
  - Binary diffing
  - Reversing firmware
  - Finding vulnerabilities
  - Short introduction to hardware reverse-engineering