Advanced Assembly & Exploitation

Course 6848 – 40 Hours

Overview

Assembly is the foundation for all kinds of applications, from mobile to desktop. This course is the full collection of x86 (32/64 bits) Assembly Adventures. It covers everything from the real basics to being an independent (and tough) Assembly programmer.
With that assembly knowledge we can create very small and powerful shellcodes, and learn how to inject through buffer/heap overflows while bypassing several security mechanisms (Linux/Windows).

On Completion, Delegates will be able to

- Program simple to advanced applications in ASM for 32/64 platforms
- Program tiny and powerful shellcodes
- Identify source code flaws for exploitation
- Inject code while bypassing security mechanisms (compiler & OS)

Who Should Attend

- Low-Level Developers
- Researchers
- Hands-on Pentesters
- Malware Analysts

Prerequisites

- Advanced C programming
- Intro to basic Operating Systems mechanisms (Linux & Windows)

Course Contents

Introduction

- Numeral systems (Oct, Hex, Dec, Bin), 2's complement, divisibility, boolean functions (OR, AND, NOR, NAND, XOR), boolean algebra, DeMorgan's Theorems, Karnaugh maps, Endianness.
- C language recap - I/O, arrays, structs, unions, pointers, memory management, arithmetic, strings.
- Debugging with gdb / VS, symbol tables
- Compiling & Linking - static vs dynamic linking

CPU Design & Architecture

- Segmentation & Virtual Different archs overview, RISC / CISC, Intel x86 and AMD evolution & overview & history
- Architecture of a very simple microcomputer and introduction to assembly language.
- Architecture of the Intel 8086, ASCII Code and 8086 assembly language environment.
- Memory and privilege levels
- User / Kernel mode

Assembly

- x86, x86-64, ARM, MIPS overview
- AT&T vs. Intel Syntax
- Assembler intro
- Data types
- Conditionals, branching
- x86 registers,x86-64 registers, flags, operand types (Intermediate, Register, Memory)
- Arithmetic Operations & expressions
- Functions prologue and epilogue, stack operations & call stack, calling conventions, recursion
- Floating point operations
- Addressing modes

Exploitation

- Disassembly & debugging with objdump & gdb& ollydbg
- PE / ELF / Mach-O - structures, sections, import/export tables, PLT/GOT, resources, etc.
- Stack exploitation
- Stack protections & How to bypass
    - Stack Canary
    - NX-bit (DEP)
        - Return-to-libc attack
        - ROP & Gadgets
    - ASLR
        - Return-to-plt
        - libc address leakage
- Heap management and exploitation
    - Heap overflow
    - Heap spray
    - UAF - Use after free
- Anti-debugging techniques