Module A1: x86 and x64 architectures and assembly languages

**Module Description:** This module will provide an overview of the major architectural features of x86 and x64 platforms, including registers, privilege levels, memory management, the associated assembly languages, the fundamental differences between x86 and x64, and an introduction to the Intel manuals (as an authoritative source that can provide vital insight during the SRE process).

**Prerequisite Knowledge:** This is intended to be a review as it is expected that students in an SRE class will have previously taken an architecture or assembly language class, although possibly not an x86 or x64 specific architecture class.

**Length of Completion**: Module – More than 4 and less than 10 hours

**Level of Instruction:** This module intended to be an upper division undergraduate or graduate course for technical majors.

**Learning Setting:** This module is intended for in-class.

**Lab Environment:** VMs and containers that include the exercises and challenge materials are available to ensure consistency and ease of deployment.

**Lab Tasks:** The objective of this lab is to demonstrate mastery of the prerequisite material by extending the assembly language concepts discussed in class to create a new functional product that meets the student’s self-defined requirements.

**Lab Files Needed:** All files are contained in the VMs.

# learning outcomes

MODULE learning oUTCOMES

Upon successful completion of this module, the student should be able to:

1. Analyze the major architectural features of x86 and x64 platforms, including registers, privilege levels, and memory management.
2. Compare and assess the fundamental differences between x86 and x64.
3. Identify and navigate the Intel manuals contents as an authoritative source that can provide vital insight during the SRE process.
4. Differentiate (compare, infer, annotate) simple assembly language programs.
5. Design and implement assembly language programs for x86 and x64

# module Details

**Interconnection:** This module is part of a 15-week Software Reverse Engineering (SRE) Course. The 15 modules are:

* **Module A1: x86 and x64 architectures and assembly languages**
* Module A2: ARM architectures and assembly languages
* Module A3: "Forward Engineering"
* Module B1: Approaches
* Module C1: Techniques for the safe handling of files of unknown origin and /or functionality
* Module C2: Basic static analysis tools.
* Module C3: Disassemblers and Decompilers
* Module D1: Sandboxing and other techniques for the safe execution/opening of files of unknown origin and /or functionality.
* Module D2: Basic dynamic analysis tools.
* Module D3: Debuggers
* Module D4: Network Traffic Analysis
* Module D5: Patching binaries
* Module E1: Obfuscation and Anti-SRE
* Module F1: Non-Binary SRE

**Instructional Files and Online Resources Needed:**

PowerPoint file: SRE-A1-Lecture.pptx

Lab Environment: SRELNX-2.ova (Format may vary)

Lab Narrative: SRE-A1-Lab.docx

Supplementary Reading:

Sikorski, M. & Honig, A. (2012). *Practical Malware Analysis: A Hands-On Guide to Dissecting Malicious Software.* San Francisco:No Starch Press.

Eagle, C. (2011). *The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler (2nd ed.).* San Francisco:No Starch Press.

**Assessment:**

The learning objectives are assessed through the following methods:

1. ASKs: There are questions in the note sections for the instructor to involve the students in the lesson and assess their grasp of the concepts.
2. LABs: There are lab exercises associated with this lesson.

# lesson

**Warm Up:** See SRE-A1-Lecture.pptx

**Lesson:** See SRE-A1-Lecture.pptx

**Active Learning Activity:** See SRE-A1-Lab.docx