Module E1: obfuscation and anti-sre.

**Module Description:** Reverse engineers often have to perform SRE on binaries that have been specifically designed or modified in order to make the SRE process more challenging. Malware in particular often includes such obfuscation and anti-SRE techniques, although many legitimate programs also include similar techniques (e.g., computer games). Common techniques include the obfuscation of instruction sequences, packing the binary, execution virtual machines (such as *VMProtect*), checksums or timing checks on critical code sections, anti-debugging techniques, VM detection, and sandbox detection. This module will cover these techniques, including methods by which they can be bypassed, eliminated, and mitigated by an analyst.

**Prerequisite Knowledge:** This is intended to be a follow-up to Module D5: Patching binaries.

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

**Level of Instruction:** This module intended to be an 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 to ensure consistency and ease of deployment.

**Activity/Lab Tasks:** A series of hands-on exercises with associated materials to guide instructors and/or students through the process

* Hands-on exercises and challengesduring this module include demonstrating anti-SRE techniques as well as circumventing anti-SRE techniques.

**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 evaluate and utilize the following techniques for making SRE more challenging:

* + Introduction
  + Debugger Detection
  + Code Verification
  + Code Timing
  + VM detection
  + Packed/Dynamic Code
  + Sandbox detection

# 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 that are Needed:**

PowerPoint file: SRE-E1-Lecture.pptx

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

Lab Narrative: SRE-E1-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

# lessons

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

**Lesson:** See SRE-E1-Lecture.pptx

**Active Learning Activity:** See SRE-E1-Lab.docx