Module F1: Non-binary sre.

**Module Description:** It is easy to think of malicious files as just binaries, but in the modern computing environment there are many other vectors for an adversary to introduce malicious content. Many common file formats (including Word, Excel, PDF, etc.) including the capability to embed code (such as VBA or JavaScript) within the file, or to reach out to pull content from external network sources. In addition, the modern web is essentially driven by code (JavaScript typically) being delivered over the network and executed in the browser. All of these are vectors for malware delivery, and are often highly obfuscated. In this module, the focus is on the application of SRE approaches and techniques (static and dynamic) to the challenge of understanding what functionality is included in web pages and non-executable files.

**Prerequisite Knowledge:** This is intended to be a follow-up to Module E1: Obfuscation and Anti-SRE.

**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.

**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 will include the opportunity to investigate a non-binary SRE tool (or an RE tool that provides anti-RE functionality.)

**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:

* + Describe and discuss other types of reverse engineering beyond compiled binaries.
  + Explain how embedded functionality can be found in a variety of formats that may seem benign at a superficial level.

# 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-F1-Lecture.pptx

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

Lab Narrative: SRE-F1-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-F1-Lecture.pptx

**Lesson:** See SRE-F1-Lecture.pptx

**Active Learning Activity:** See SRE-F1-Lab.docx