

## Software Reverse Engineering Course

This course aims to educate students in the main usage scenarios and contexts in which software reverse engineering (SRE), or reversing, is used. In addition, the course will provide students with the opportunity to gain facility with current reverse engineering software tools and techniques through hands-on exercises. Several scenarios have been developed (complete with descriptive scenarios) and each includes a series of exercises, ranging from easy to very difficult.

Students will learn that there are two main categories of software reverse engineering applications: development-related and security-related. Four mainline usage scenarios will be covered within each of the two categories.

- Development-related: achieving interoperability with proprietary software, verification that an implementation matches the design, evaluating software quality and robustness, and legacy software maintenance, re-engineering, and evolution.
- Security-related: detecting and neutralizing viruses and malware, testing digital rights management (DRM) software protection, auditing the security of program binaries when the source is unavailable, and testing cryptographic algorithms for weaknesses.

Based on the above usage scenarios, three modules will be taught that represent the most relevant concepts (some of the scenarios listed above will be lecture-only). Students will receive instruction and complete hands-on exercises in each of the following.

- Software Reversing, Antireversing, and Patching
- Design Recovery and Software Reuse
- Software Security and Malware

The expected learning outcomes for students of the course are:

- An improved ability to understand, maintain, evolve, and secure software.
- An understanding of the most popular and relevant uses of software reversing.
- Facility with several software reverse engineering tools.
- Facility with assembly language and interpreted languages like Java Bytecode.