Motivation

Our technological society has become more and more dependent on software that is used to automate everyday processes. This dependence increasingly exposes us to security threats that originate from malicious software (malware) such as computer viruses and worms and software vulnerability exploits such as remote execution of code or denial of service attacks. Moreover, this exposure is not limited to computer systems but is spreading to common appliances such as mobile phones, PDAs and consumer electronics such as media centers, personal video recorders, etc. since a growing number of these products are made extensible and adaptable by means of embedded software.

The proliferation of malware and exploits requires that action is taken to tackle these issues and better evaluate software security to prevent the damage and costs (e.g., data loss, productivity loss, recovery time) that result from security incidents. This calls for measures to assure that a software system has the desired security properties, i.e. that it is free of malware and vulnerabilities. Besides attention for techniques that aim at secure development of new systems, there is a strong need to investigate measures that can deal with the large amount of existing (legacy) code.

Objectives

The purpose of this workshop is to bring together practitioners, researchers, academics, and students to discuss the state-of-the-art of software security assessments based on reverse engineering of source or binary code (as opposed to software security assessments that look at the software process that was applied). This includes research on topics like source & binary code analysis techniques for the detection of software vulnerabilities (e.g. detect if code has potential buffer overflow problems) or analysis for the detection of malicious behavior (e.g. detect if code contains an exploit or has viral behavior).

The goal is to share experiences, consolidate successful techniques, collect guidelines, and identify opportunities for future work and collaboration. We will do so by building on the list of open issues in software security research collected, refined and prioritized by participants of CoBaSSA 2005.

Topics of interest

CoBaSSA topics of interest include, but are not limited to:
- Mitigating stack- or heap-based buffer overflow attacks
- Re-modularizing legacy code for privilege separation
- Race condition detection
- Vulnerabilities in trust management and authentication
- Code tamper-proofing and obfuscation
- Decompilation, disassembly and anti-debugging
- Copy protection schemes
- Analysis of computer virus and worm code
- Case studies in analyzing software vulnerability
- Best practices practices for secure coding

Submissions & Workshop Format

Participants are asked to submit a four page position paper detailing their experiences or ideas on software security assessments. The organizers accept position statements based on originality, relevance, and suitability for triggering discussion. All accepted papers are published on the workshop web-site.

The workshop aims at discussion and interaction rather than presentations. However, participants will have the chance to give a short presentation which typically serves to introduce a case study, provoke discussion by presenting a controversial point of view, or introduce new points of view. In order to stimulate debate, each position paper will have a discussant assigned, who has the task to study the position paper in advance, and prepare one or two questions.

The workshop opens with an introduction session where participants can raise the questions they would like to get addressed in the workshop; in the concluding wrap up we’ll evaluate how far we got in answering these questions.

Further information is available on the workshop web-site.