As software testing becomes a more complex and extensive process, an opportunity exists to bring together researchers and practitioners from a variety of testing venues to share promising methods, research and technologies. This minitrack focuses on software testing in general and internet testbeds in particular. The papers presented here reflect the variety, scope, and scale of testing ranging from embedded systems testing to large scale network emulation testing. Each paper offers a unique or new way to approach the problem of assuring that the systems perform only their intended functionality and do not include any new vulnerabilities or unexpected outputs.

The first paper, “New Trends in Security Evaluation of Bayesian Network based Malware Detection Models,” by Eric Filiol and Sébastien Josse discusses the design and evaluation of statistical information retrieval models, presents information-theory-based criteria to characterize the effectiveness of spectral analysis models, and discusses the limits of such models.

The second paper, “Cloud Chamber: A Self-Organizing Facility to Create, Exercise, and Examine Software as Service Tenants,” by M. Brent Reynolds, Donald Hulce, Kenneth Hopkinson, Mark Oxley, and Barry Mullins, presents a testbed for understanding how web services new behavior as tenants in a Software as a Service environment. The testbed inserts sensors into web servers to collect performance data and generates profiles of resource usage for services and availability of servers. The information is used to calculate configurations which better meet changing requirements.

The third paper, “Effort Estimates for Vulnerability Discovery Projects,” by Teodor Sommestad, Hannes Holm and Mathias Ekstedt, analyzes weighted estimates from domain experts using Cooke’s classical method on the amount of effort required for a penetration tester to find zero-day vulnerability in a software product.

The fifth paper, “On the Fault-Detection capabilities of Adaptive Random Test Case Prioritization: Case Studies with Large Test Suites,” by Zhi Quan Zhou, Arnaldo Sinaga, and Willy Susilo, the authors investigate the fault-detection capabilities of using frequency information for adaptive random test case prioritization and then comparing Jaccard Distance and Coverage Manhattan Distance. Their results show which approach is superior and how they can be used in a complementary fashion.

The fifth paper, “Effectiveness of Random testing of Embedded Systems,” by Padmanabhan Krishnan, R. Venkatesh, Prasad Bokil, Tuakaram Muske, and Vijay Suman, examines how practitioners can choose an effective technique to test their systems in an embedded environment. The authors present their findings about test case generation at both the system and unit testing levels.

In the sixth paper, “Partitioning Trust in Network Testbeds,” by Gary Won, Robert Rici, Jonathon Duerig, Leigh Stroller, Srikanth, Chikkulapelly, and Woojn Seok, the authors argue that partitioned trust is increasingly important in large-scale and security-sensitive testbeds. They present a design that accomplishes partitioning by using multiple trust roots. They explain the details of their implementation and share experiences of using it with hundreds of users.

The seventh paper, “Automated Behavior Computation for Software Analysis and Validation,” by Mark Pleszkoch, Richard Linger, Stacy Prowell, Kirk Sayre, and Luanne Burns, describes the emerging technology of software behavior computation as a means to derive the full functional effect of software for validation of functionality and analysis of security properties. A system to automate this process, the Function Extraction system, is described, with special focus on implementation of loop behavior computation. Use of a behavior computation system is illustrated for validating a miniature looping program with and without embedded malware.