

ConfigAssure: Dynamic System Configuration Assurance for National Intelligence Communities Cyber Infrastructure

Sanjai Narain¹, Telcordia, Daniel Jackson, MIT, Sharad Malik, Princeton

What are you trying to do? We are trying to eliminate vulnerabilities in cyber infrastructure due to configuration errors. Configuration errors are exploited in 65% of cyber attacks and are the cause of 35% of infrastructure downtime. We are also trying to reduce the time taken to design assured information sharing infrastructure by one order of magnitude.

How is it done at present? Who does it? What are the limitations of the present approaches? There is a large conceptual gap between end-to-end security requirements and detailed component configuration implementing those requirements. Today, this gap is *manually* bridged. There are no tools to precisely specify end-to-end security requirements, to automatically compile these into component configurations, to diagnose and fix configuration errors, and to safely change the current configuration to the desired one in real-time. Compounding the challenge is the inherent tension between security and functionality: security is about preventing bad behavior whereas functionality is about enabling good behavior. Incorrect resolution of this tension can disable mission-critical services and potentially cause as much harm as allowing an adversary access to those services. The configuration search space in a typical infrastructure is very large arising out of 100s of requirements, 10s of protocols, 100s of components each with 100s of configuration parameters and possible values. For these reasons, a cyber infrastructure often contains a large number of configuration errors and also takes weeks or months to set up.

What is new about your approach? Why do you think it will be successful? The novelty in our approach is exploiting the power of modern SAT-based model finders to bridge the above gap. Boolean SAT solvers can solve millions of constraints in millions of variables in seconds. Security requirements are expressed as constraints on configuration parameters that are then solved to compute their values. By representing both security and functionality requirements as constraints, the tension between these is automatically resolved. If constraints are unsolvable, the proof of unsolvability provides a systematic method of diagnosing and fixing configuration errors. Safe transition into the correct configuration is also expressible as a constraint satisfaction problem. Model-finders allow constraint specification in an intuitive and expressive first-order logic language, compile these into Boolean logic, solve these with a SAT solver, then reflect results back into first-order logic. We think our approach will be successful because we have developed a method of suppressing generation of very large intermediate constraints in the translation of first-order logic into Boolean. The idea is to “factor away” subsets of constraints that can be solved via specialized constraint solvers, leaving behind a constraint that truly requires the power of model finding via SAT. A new “deductive spreadsheet” user-interface simplifies use by infrastructure administrators.

If you succeed, what difference will it make? We expect to eliminate vulnerabilities in cyber infrastructure due to configuration errors and reduce the time taken to design assured information sharing infrastructure by one order of magnitude.

How long will it take? How much will it cost? What are your mid-term and final exams? The project duration is 18 months ending in December 2008. The total cost is \$956K. By project end, we plan to complete the design and implementation of algorithms to solve above fundamental configuration problems. Our mid-term exam has been solving these for a real, secure and fault-tolerant data center at Telcordia. Our final exam will be solving these for a DoD collaboration infrastructure.

¹ Email: narain@research.telcordia.com, Tel: 732 699 2806, Cell: 908 337 3636