DARPA's HACMS Program: A Proof of Concept of Formal Methods at Scale[1]

Kathleen Fisher, Tufts University

DARPA's HACMS program demonstrated that systems built using formal methods could be significantly more secure than current norms. HACMS researchers focused on a pair of platforms: an open-source quadcopter accessible to all researchers and Boeing's Unmanned Little Bird (ULB) helicopter, accessible only to Boeing engineers. This structure created a workflow in which researchers developed tools and techniques and demonstrated them on the quadcopter, and then transition vehicle experts applied the tools and techniques to the ULB.

A team of professional penetration testing experts, a "Red Team" assessed the security of the vehicles at the beginning of the program and then again at the end of each of the programs' three 18-month phases. The Red Team had full knowledge of the vehicles: they had access to all the relevant documentation and source code, and they participated in the various design meetings. At the start of the program, the Red Team easily took remote control of the quadcopter and the ULB. In all subsequent tests, however, the Red Team was not able to disrupt the operation of the vehicles re-engineered with formal methods, even when they were allowed to run code of their own devising with root access in a legacy partition.

To achieve these results, the formal methods researchers re-architected the software running on the vehicles using a variety of formal methods techniques. The low-level C code that ran on the flight-control computer was generated from domain-specific languages along with proof obligations. The mission-control computer, which handled communication with the ground station and directed the flight-control computer, ran Data61's formally-verified seL4 microkernel, configured to have two partitions. The first partition contained all the security-critical code, including the code for communicating with the flight-control computer and the ground station. The second partition ran non-security-critical code, specifically the vehicle's camera software on top of a Linux installation (It was in this second partition that the Red Team installed their code). HACMS researchers proved a number of system-wide security properties, including memory safety, protection against malformed or non-authenticated messages, and a guarantee that authenticated messages received by the vehicle would be acted upon.

It is hard to overstate the significance of these results, which demonstrated a way to dramatically improve the security of an existing system by leveraging high-assurance artifacts, such as the verified seL4 microkernel, and reimplementing security-critical code using formal methods. Critically, not all of the software had to be re-built. Non-security critical code was isolated in "legacy" partitions. This approach provides a way to "cyber-retrofit" an existing system, analogous to seismic retrofits that improve the earthquake safety of existing buildings.

In the final phase of the HACMS program, researchers began to work with a variety of other vehicles and transition partners to demonstrate that the results were not particular to the quadcopter and the Boeing ULB. The success of the HACMS project led to a clause in the John McCain National Defense Authorization Act, Section 1657(c)(3), directing the DoD to study "Formal programming and protocol language for software code development and other methods and tools developed under various programs such as the HACMS program."

---

[1] For more information about the HACMS program, see *The HACMS Program: Using Formal Methods to Eliminate Exploitable Bugs*, Philosophical Transactions A, 375(2104), 2017.